

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
19 June 2003 (19.06.2003)

PCT

(10) International Publication Number  
**WO 03/050744 A1**

(51) International Patent Classification<sup>7</sup>: **G06F 17/60**,  
3/00, 13/38

(21) International Application Number: PCT/US02/39288

(22) International Filing Date: 6 December 2002 (06.12.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/336,988 7 December 2001 (07.12.2001) US  
60/337,793 10 December 2001 (10.12.2001) US

(71) Applicant (*for all designated States except US*): **SOFC-CAST, INC.** [US/US]; 36 Blakeley, Irvine, CA 92620 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— with international search report  
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

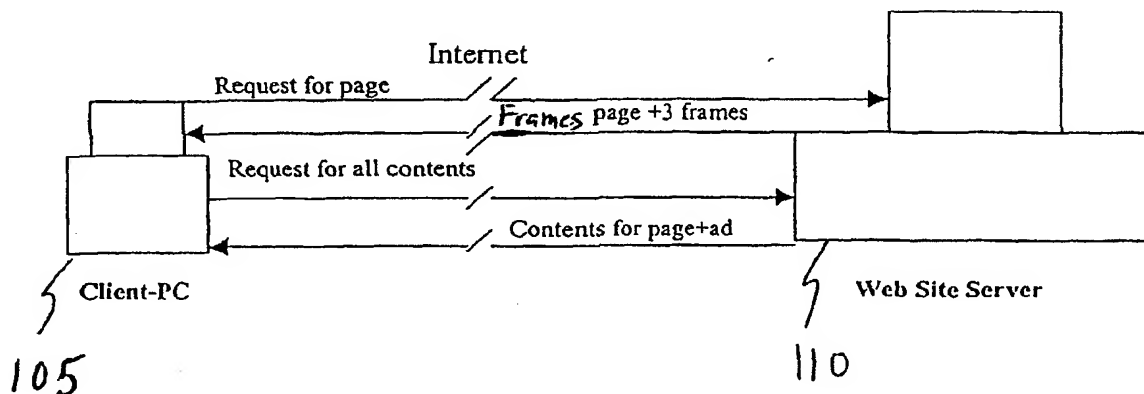
(72) Inventor; and

(75) Inventor/Applicant (*for US only*): **KOLSY, Mohammed, H.** [US/US]; 36 Blakeley, Irvine, CA 92620 (US).

(74) Agents: **KAPOUYTIAN, Ararat et al.**; Morrison & Forster LLP, 425 Market Street, San Francisco, CA 94105-2482 (US).

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

(54) Title: DELIVERING CONTENT AND ADVERTISEMENT



(57) Abstract: A method and apparatus for delivering content and advertisement is described. In one embodiment, the method includes: receiving a request for a target web page; sending a frames page including a plurality of frames and a frames page program module (410), where the plurality of frames comprises a content frame and an advertisement frame; receiving a request to load the content frame (415); sending content to load the content frame (412), where the content includes the target web page (412); receiving a request to load the advertisement frame (413). In another embodiment, the method includes: receiving a request for a target web page; receiving a request to load the content frame; sending content to load the frame; receiving a request to load the advertisement frame and sending an advertisement to load the advertisement frame.

WO 03/050744 A1

## DELIVERING CONTENT AND ADVERTISEMENT

Inventor: Mohammed H. Kolsy

### CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefits of U.S. Provisional Application Serial Numbers 60/336,988 and 60/337,793, filed December 7, 2001, and December 10, 2001, respectively, and entitled "Technology for Delivering Web Pages and Ad Messages in Split Screen Configuration Without Programming" and "Dual Frame Technology for Interstitial Advertising/Messaging on the Internet or Any Network Environment," respectively, both of which are incorporated herein by reference.

### BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

[0002] This invention relates generally to delivery of information and, in particular, to delivery of primary content and advertisement.

#### 2. Description of the Related Art

[0003] Many conventional methods of advertisement delivery over a network, such as the Internet, involve web pages which are coded to account for advertisements associated with the pages. In some methods, the advertisement is displayed as part of the web page. In such a case, the advertisement has a predetermined assignment of size and location in the displayed web page. This predetermined assignment is part of the coding for the web page. Modifying this predetermined assignment requires recoding of the web page.

[0004] In some methods, the advertisement is displayed during the interstitial time between display of web pages. In such methods, the web page includes tag lines that send requests for retrieving the associated advertisement from the proper server before displaying the subsequent web page. In such a method, when a user clicks on

a hyperlink for next page the tagline for advertisement gets activated and is downloaded first by the client before the requested web page is retrieved. This creates a delay in receiving the advertisement at the client. Moreover, it does not provide for modifying the association between the web page and the advertisement without recoding the web page as the association is part of the coding for the web page. In addition, these prior art methods are capable of delivering only one type of advertisement display system.

[0005] In all of the above methods, recoding or reprogramming of the web page is required to modify the predetermined relationships between the web page and the advertisement.

[0006] The present invention addresses this and other disadvantages of existing methods.

#### BRIEF SUMMARY OF THE INVENTION

[0007] The present invention encompasses a method of delivering content and advertisement. In one embodiment, the method includes: receiving a request for a target web page; sending, in response to the request for the target web page, a frames page including a plurality of frames and a frames page program module, where the plurality of frames comprises a content frame and an advertisement frame; receiving a request to load the content frame; sending content to load the content frame, where the content includes the target web page; receiving a request to load the advertisement frame; and sending an advertisement to load the advertisement frame.

[0008] In another embodiment, the method includes: receiving a request for a target web page; sending, in response to the request for the target web page, a web page that causes, directly or indirectly, establishing a frames page including a plurality of frames and a frames page program module, where the plurality of frames includes a content frame and an advertisement frame; receiving a request to load the content frame, where the request to load the content frame is prompted by the frames

page program module; sending content to load the content frame, where the content includes the target web page; receiving a request to load the advertisement frame, where the request to load the advertisement frame is prompted by the frames page program module; and sending an advertisement to load the advertisement frame.

[0009] The present invention is explained in more detail below with reference to the drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Figure 1 is a block diagram of a client and server and the general interaction therebetween according to one embodiment of the present invention.

[0011] Figure 2 is a block diagram of major software components on the client side in one embodiment of the present invention.

[0012] Figure 3 is a block diagram of major software components on the server side in one embodiment of the present invention.

[0013] Figure 4 is an interaction diagram for browsing a web site in accordance with one embodiment of the method of the present invention.

[0014] Figure 5 is an interaction diagram for navigating a web site in accordance with one embodiment of the method of the present invention.

[0015] Figure 6 is an interaction diagram for clicking on an advertisement in accordance with one embodiment of the method of the present invention.

[0016] Figure 7 is an interaction diagram for exiting a web site in accordance with one embodiment of the method of the present invention.

[0017] Figure 8 shows a flowchart of major functions of the frames page builder script (index.php) in one embodiment of the present invention.

[0018] Figure 9 shows a flowchart of major functions of the content caching script (adload.php) in one embodiment of the present invention.

[0019] Figure 10 shows a flowchart of major functions of the advertisement display script flow chart (advview.php) in one embodiment of the present invention.



## DETAILED DESCRIPTION OF THE INVENTION

[0020] The present invention comprises a method of delivering content and advertisement. The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the embodiments shown will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[0021] The present invention is described in relation to the delivery of content and advertisement. However, it is to be noted that the present invention is not limited to the delivery of content and advertisement. Instead, it is broadly applicable to the delivery of first and second types of information. For example, it is applicable to the delivery of a first primary content and a second primary content.

[0022] Figure 1 is a block diagram of a client and server and the general interaction therebetween according to one embodiment of the present invention. In Figure 1, client 105 is coupled to server 110. In one embodiment, client 105 is a computer, while server 110 is a web server, and client 105 is coupled to server 110 via the Internet. The server 110 includes the server-side software of the present invention. In another embodiment, client 105 may be coupled to server 110 via an intranet or some other type of network.

[0023] Figure 1 also shows, at a general level, the interaction between the client 105 and the server 110 in one embodiment of the present invention. The client 105 sends a request for a web page or web site to the server 110. The server 110 receives the request. If during the current session between the client 105 and server 110, the request is a first request from client 105 for the web page or web site, then the server 110 sends a frames page (also herein referred to as an index page or holder

page) and a plurality of empty frames to the client 105. More specifically, a frames page, among other things, defining a plurality of empty frames is sent to the client 105. In one embodiment, the page sent by the server 110 in response to the first user request can be a JavaScript script or a reference to a program that when executed by the browser effectuates the same result, i.e., establishing in the client browser a frames page and the plurality of empty frames.

[0024] In one embodiment, if the request has no frames page identification and is a content request, then it is determined to be a first request. In other words, if the request Universal Resource Locator (URL) refers directly to a content web page rather than an ad server component, then it is a first request. The client 105 stores the frames page and plurality of empty frames in the browser cache. Once the frames page is received at the client 105, it supersedes some of the browser functionality, particularly those relating to the user interface, at the client 105.

[0025] In one embodiment, two empty frames are sent from the server 110 to the client 105. One empty frame is a content frame for loading the requested web page. Another empty frame is an advertisement frame for loading an advertisement. In another embodiment, three empty frames are sent from the server 110 to the client 105. The three empty frames include a content frame, an advertisement frame, and a daemon frame. The daemon frame is for storing a record of the browsing activity during the session.

[0026] The frames page includes part of the client side software used in performing the method of the present invention. Figure 2 is a block diagram of major software components on the client side in one embodiment of the present invention. As shown in Figure 2, major software components on the client side include a frames page program module 205, a content page program module 210, and an advertisement page program module 215.

[0027] As shown in Figure 2, the frames page program module 205 includes software for performing the following functions: (1) handling hyperlink mouse click

actions; (2) loading content; (3) loading advertisements; and (4) allocating screen area. In one embodiment, the frames page program module 205 includes programming code, such as JavaScript scripts, for performing these functions. In one embodiment, the JavaScript scripts are used with Hypertext Markup Language (HTML) codes. Appendix 1 shows an example of the JavaScript scripts included in the frames page program module 205. It is to be noted that the frames page program module 205 can be written in other languages besides JavaScript (e.g., JScript or other non-Java based languages).

[0028] Handling the hyperlink mouse click actions involves monitoring for and responding to hyperlink mouse click actions. Loading content involves sending or prompting the sending of a request for the content of a web page, such as the web page previously requested by the client 105. Similarly, loading the advertisement message involves sending or prompting the sending of a request for the advertisement message and loading the advertisement message into the advertisement frame. Finally, allocating screen area involves splitting the screen area allocated to the browser between the content frame and the advertisement frame. It also involves adjusting or reallocating the distribution of the screen area between the content and advertisement frames. As used herein splitting includes allocating screen area such that one of the frames has only a 0% of the screen area. For example, it includes making a 0% and 100% allocation of the screen area between the content frame and the advertisement frame.

[0029] As noted above, once the frames page is received at the client 105, it supersedes some of the browser functionality at the client 105. Under control of the frames page, more specifically the script in the frames page program module 205 related to the function of loading content, the client 105 sends a request for the content of the web page previously requested by the client 105. In response, the server 110 retrieves the requested content and adds to the content page scripts for performing the following functions: (1) handling mouse click events; (2) handling

window OnLoad events; and (3) handling window OnBeforeUnload events. These scripts may herein be referred to as scripts in the content page program module 210. In one embodiment, the server 110 modifies the content page by rewriting URLs in the content page such that they include a reference to the server 110, more specifically the adext.dll module of the server 110. Each rewritten URL references a web page via the server 110, more specifically the adext.dll module of the server 110, rather than referencing the web page directly. In another embodiment, rather than modifying the content page by rewriting the URLs therein, the server 110 adds a one line JavaScript script to the content page. This one line JavaScript script intercepts hyperlink clicks on the content web page by the user. It also dynamically rewrites the URL corresponding to the hyperlink such that the user click is redirected to the server 110. Accordingly, the requests represented by these hyperlink clicks are then forwarded to the server 110. In one embodiment, an advertisement identification (ID) is added to the content page. The advertisement ID identifies an advertisement associated with the content page. Thereafter, the server 110 sends the content page with the scripts of the content page program module 210 to the client 105.

[0030] As shown in Figure 2, the content page program module 210 includes instructions for performing the functions of: (1) handling mouse click events; (2) handling window OnLoad events; and (3) handling window OnBeforeUnload events. Appendix 2 shows an example of the JavaScript scripts included in the content page program module 210. It is to be noted that the code in the content page program module 210 can be written in other languages besides JavaScript (e.g., JScript or other non-Java based languages).

[0031] Handling mouse click events involves monitoring for and responding to mouse click events. If the mouse click event involves a hyperlink click event, then the content page program module 210 notifies the frames page program module 205 of the event. Handling window OnLoad events involves three important functions: initiating the retrieval of subsequent content ( using Ad ID, ad page is retrieved in this

case), "content loaded" flag is set, and certain conditions are checked with configurations in the frames page to determine whether to load the next advertisement. In one embodiment, the performance of these functions involves notifying the appropriate scripts in the frames pages program module 205. Similarly, handling window OnBeforeUnload events involves handling events prior to the unloading of windows. An example of such an event includes setting up a number of global variables and LoadAd function is called in the holder scripts. When content is unloaded from a content frame in the window, the content page program module notifies the frames page program module of the unloading. Thereafter, the frames page program module displays an advertisement in the advertisement frame on the window by resizing the content frame to 0% of the screen area and the advertisement frame to 100% of the screen area for interstitial display.

[0032] The client 105 receives the content page, which includes the content page program module 210. Under the direction of the scripts for loading content, the empty content frame is filled with the content page. Thereafter, the content page is displayed on the screen at the client 105.

[0033] In one embodiment, upon receiving the content page, per the instructions in the frames page, more specifically, the scripts for loading the advertisement message, the client 105 sends a request to the server 110 for the advertisement. The server 110 retrieves the advertisement page and adds to the advertisement page scripts for performing the following functions: (1) displaying buttons, such as "HOLD" and "HOME," and (2) handling button click events for the "HOLD" and "HOME" buttons. The "HOME" button may also herein be referred to as a "LEARN MORE" button. Thereafter, the advertisement page with the above mentioned scripts is sent to the client 105.

[0034] As shown in Figure 2, the advertisement page program module 215 includes instructions for performing the functions of: (1) displaying buttons, such as "HOLD" and "HOME" buttons, and (2) handling button click events for the

“HOLD” and “HOME” buttons. Appendix 3 shows an example of the JavaScript scripts included in the advertisement page program module 215. It is to be noted that the code in the content page program module 210 can be written in other languages besides JavaScript (e.g., JScript or other non-Java based languages).

[0035] When a user clicks on a “HOLD” button associated with an advertisement, the advertisement page program module 215 captures the “HOLD” button click event. Thereafter, the advertisement page program module 215 notifies the frames page program module 205 of the “HOLD” button click event. The frames page program module 205 continues to display the advertisement even if the content page with which the advertisement is associated is no longer displayed on the window. In one embodiment, a user is able to surf through the web site while a video clip advertisement continues to play on the screen in split screen mode (which is described in greater detail below). When a user clicks on a “HOME” button associated with an advertisement, the advertisement page program module 215 captures the “HOME” button click event. Thereafter, the advertisement page program module 215 notifies the frames page program module 205 of the “HOME” button click event. The frames page program module 205 opens a new browser window in which the home page of the advertiser is displayed. In one embodiment, the URL of the advertiser’s home page is stored in the database 340 (shown in Figure 3) and maintained using the administrative ASP pages 310 (also shown in Figure 3). The frames page program module 205 retrieves this URL from the database 340.

[0036] In one embodiment, an advertisement that is associated with the content page is retrieved and sent to the client 105. In one embodiment, the advertisement is selected based on the advertisement ID mentioned above. In one embodiment, the advertisement is randomly selected from a list of advertisements that are associated with the content page.

[0037] In one embodiment, the advertisements are stored as HTML files on an advertisement server which may be server 110 or some other server. The HTML

files may be rewritten by the Display Script to insure that HTML tags and attributes for image, flash, video, or other content refer to the correct URL within the web site.

[0038] The client receives the advertisement page with the above mentioned scripts. Under the direction of scripts in the frames page for loading the advertisement message, the advertisement frame is filled with the advertisement page.

[0039] There are two general embodiments for displaying of the advertisement page in the present invention. In one embodiment, the advertisement page is displayed on the screen concurrently with the content page in multiple frames. Such an embodiment is herein referred to as a split screen mode of the present invention. In another embodiment, the advertisement page is displayed only between, in terms of time, the display of successive content pages. In other words, it is

---

displayed during the interstitial time period between displaying content pages. Such an embodiment may herein be referred to as an interstitial mode embodiment of the present invention. In the interstitial mode embodiment, the loaded advertisement page is stored in the browser cache while the web page content is displayed.

[0040] In one embodiment, the content frame and the advertisement frame are allocated 100% and 0% of the screen area (more specifically, the area allocated to the browser window), respectively. This allocation is maintained while the content page is active. Once the user initiates a request to retrieve another web page, either within or outside the web site to which the current web page belongs, the scripts for allocating screen area adjust the allocation of display space to the content and advertisement frames. With this adjustment, the content frame and advertisement frame are allocated 0% and 100% of the screen, respectively. Consequently, the advertisement frame with the advertisement page therein is displayed on the screen at client 105. The advertisement page is displayed relatively quickly because, as noted above, prior to the adjustment in allocation it was already preloaded in the background in browser cache.

[0041] In one embodiment, the advertisement page is displayed until the next requested page is received by the client 105. The next requested page replaces the content page. Thereafter, the allocation to the content frame and advertisement frame is readjusted to 100% and 0%, respectively. In another embodiment, the advertisement frame with the advertisement page therein is displayed until an advertisement duration timer runs out.

[0042] The above-described embodiments implement a time-division multiplexing aspect of the present invention wherein the available screen space is entirely devoted at any one time to the display of either the content page or the advertisement page.

[0043] In another embodiment, the content frame and the advertisement frame are both allocated a non-zero percentage of the screen area. In one embodiment, the content frame and the advertisement frame are allocated 80% and 20% of the screen, respectively. In another embodiment, other non-zero percentages may be allocated to the content frame and the advertisement frame. In one embodiment, the screen is divided vertically between the content and advertisement frames. In another embodiment, the screen is divided horizontally between the content and advertisement frames. In yet another embodiment, there may be a combination of horizontal and vertical division or some other type of division between the content and advertisement frames displayed on the screen.

[0044] The embodiments described in the preceding paragraph implement a bandwidth division multiplexing aspect of the present invention wherein the available screen space is devoted at any one time to the display of both the content page and the advertisement page.

[0045] In the embodiment where both the content frame and advertisement frame are allocated a non-zero percentage of the screen, the content and advertisement frames (with the content and advertisement pages therein, respectively) are displayed together. If the content of the web page does not fit within the allocated



space, e.g., 80% of the screen space, then the client would automatically add a horizontal scroll bar which a user may scroll to view the full page.

[0046] In one embodiment, as the content page is filled before the advertisement page is received at the client 105, the content page is displayed in the area allocated to the content frame, while the empty advertisement frame is displayed in the area allocated to it. Once the advertisement is received at the client 105, it is displayed in the area allocated to the advertisement frame.

[0047] In one embodiment, the daemon frame is initially loaded with a blank page. It is thereafter loaded with server-side scripted content as needed. This causes the Advertisement Server to log advertising events, such as, advertisement views, clicks, and holds. When a user clicks the Hold Advertisement button on an

---

advertisement, or clicks an advertisement to view the advertiser's web site, the JavaScript scripts in the advertisement page causes the daemon frame to load the Logger Script (logger.php) from the web site with query parameters containing the advertisement ID, user action taken, and content page being displayed. Appendix 4 shows an example of the JavaScript scripts included in the Logger Script (logger.php). The Logger Script makes a request to a server to insert the advertising event information into the tracking database for later use in generating invoices.

[0048] Figure 3 is a block diagram of major software components on the server side in one embodiment of the present invention. As shown in Figure 3, server 110 includes the following software components: system director 305, administration Application Service Provider ("ASP") pages 310, report ASP pages 315, frames processor 320, content processor 325, advertisement processor 330, advertisement hit processor 335, and database 340. Each of frames processor 320, content processor 325, advertisement processor 330, and advertisement hit processor 335 comprises software code. The software codes of these different processors may be executed on the same or different computer hardware devices.

[0049] Server 110 forwards to system director 305 incoming requests for web pages. The system director 305 qualifies the request. Appendix 9 shows codes developed ( adfilter.dll) for the System director. As part of the qualification, the system director 305 determines whether a request for a web page is the first request during the current session between the client 105 and server 110. After qualifying the request, the system director 305 forwards the request to the appropriate one of frames processor 320, content processor 325, advertisement processor 330, and advertisement hit processor 335 based on the nature of the request.

[0050] If the request is a first request, then system director 305 forwards it to frames processor 320. Frames processor 320 builds a frames page and empty frames which are sent to the client 105. The frames page and empty frames are described in greater detail above.

[0051] If, on the other hand, the request is a second or subsequent request in the current session between the client 105 and server 110, then the request is forwarded to the appropriate one of content processor 325, advertisement processor 330, and advertisement hit processor 335 based on the nature of the request. Content processor 325 handles requests for content. Content processor 325 pulls the requested content page from database 340 and adds the content page program module 210 (shown in Figure 2) thereto. It thereafter forwards the content page with the content page program module 210 to the client 105 so that it would be filled into the content frame at the client 105. In one embodiment, every time a new content page is requested by the client 105, a content page program module 210 is appended to the requested content page, and the requested content page with the content page program module is sent to the client 105.

[0052] As noted above, in one embodiment, the URLs in a content page are rewritten such that they include a reference to the server 110, more specifically the adext.dll module of the server 110. As also noted above, in another embodiment, rather than modifying the content page by rewriting the URLs therein, a one line

JavaScript script is added to the content page. As further noted above, in one embodiment, an advertisement ID is added to the content page. In one embodiment, the content processor 325 performs the above modifications/additions to the content page.

[0053] Similarly, advertisement processor 330 handles requests for advertisements. Advertisement processor 330 retrieves an advertisement from the database 340 and adds the advertisement page program module 215 (shown in Figure 2) thereto. It thereafter forwards the advertisement page with the advertisement page program module 215 to the client so that it would be filled into the advertisement frame at the client 105. In one embodiment, every time an advertisement page is sent to the client, an advertisement page program module 215 is appended to the advertisement page before it is sent to the client 105.

[0054] Appendix 5 shows an example of the JavaScript scripts included in the advertisement processor (AdExt.dll). The scripts in Appendix 5 include scripts for performing the functions of the following components in Figure 3: the system director 305, the frames processor 320, the content processor 325, the advertisement processor 330, and the advertisement hit processor 335.

[0055] The advertisement hit processor 335 tracks the behavior of the user in relation to the displayed advertisements. The advertisement hit processor 335 forwards the tracked information regarding the user's behavior to database 340.

[0056] Database 340 may include the following tables: an advertisement table, a web sites table, a web sites advertisement links table, a clients table, a license table, a configuration table, and a hits (or tracking) table. The web sites advertisement links table links content pages with advertisements. In one embodiment, a content page may have multiple associated advertisements. In one embodiment, the multiple associated advertisements are displayed sequentially while displaying the content page with which they are associated. In one embodiment, these advertisements are all shown for the same length of time. In another

embodiment, some of these advertisements are shown for a longer period of time than others. The length of time for which an advertisement is shown depends on its display weighing factor. The configuration table configures the software at the server side.

[0057] The configuration table contains global options that control the overall operation of the advertising system. The keys and values stored in this table are loaded by and used by server-side scripts. In most cases, the key name for a specific configuration item is used as the variable name by which server-side scripts can access the configuration data. In one embodiment, the present invention defines the following configuration options:

Split Screen Size	Percentage of window width allocated to Split Screen advertising.
Split Screen Mode	Set to "Y" or "Yes" to display advertising in Split Screen mode. Any other value enables interstitial mode.
ExternalAds	Set to "Y" or "Yes" to display advertising when linking to content on other web sites. Any other value causes hyperlinks to other sites to display without advertising.
EnableNetscape	Set to "Y" or "Yes" to enable advertising display on Netscape and Mozilla web browsers with version numbers 6 or higher. Any other value causes the advertising system to be disabled for all Netscape and Mozilla browsers.
EnableHold	Set to "Y" or "Yes" to enable the Hold Page button in the Advertisement Frame. When enabled and the HOLD button has been pressed, clicking hyperlinks in the Content Frame will not cause the content of the Advertisement Frame to change.

[0058] The license table contains license keys issued to customers who have purchased or are licensed to use the software of the present invention. The license data includes license type codes, licensed product codes, and the Fully Qualified Domain Name (FQDN) of the web site to which a license has been granted, or the IP address if the web site does not have a FQDN. In one embodiment, the license data is encrypted using the industry standard MD5 Hash algorithm producing a license key

which the customer can install to activate the software of the present invention on their licensed web site.

**[0059]** The client table lists clients and their associated web sites for the purpose of generating invoices for all advertising generated by that web site.

**[0060]** The web site table lists the complete URL for each page that will display advertising and assigns a page ID to each page.

**[0061]** The advertisement table lists all potential advertisements, including the client of the advertisement, the prices for impression, hold, and click-through events, the location of the advertisement HTML, the destination URL, and the duration that the advertisement should be displayed. The advertisement table is associated with the client table by client name, and with the web site table by page URL.

**[0062]** The tracking table contains information about advertising events, including advertisement impressions, advertisement holds, and advertisement clicks. The table stores the advertisement ID, the content page URL, the date and time of the event, and the user action.

**[0063]** The administration ASP pages 310 enable administration of the site. For example, the administration ASP pages allow for changing the configuration options stored in the database 340. The report ASP pages 315 allow for generating reports and delivering them over a network, e.g., the Internet. A report regarding the number of impressions per advertisement would be an example of such reports. In one embodiment, access to the administration ASP pages 310 and report ASP pages 315 is limited to responsible personnel using security methods well known in the art.

**[0064]** Figure 4 is an interaction diagram for browsing a web site in conjunction with one embodiment of the method of the present invention. In response to a user 405 browsing 406 of a web site enabled with the software of the present invention, an index program module 410 returns 411 a frames page to the client. Loading of the frames page at the client triggers the loading of the content,

advertisement, and daemon frames. More specifically, loading of the frames page triggers load requests for the content, advertisement, and daemon frames. It also triggers a request 412 to the content program module 415 to load the content frame with the contents of the web page that the user originally requested. In response to this request, the requested content is returned 416 to the user in the filled content frame. Before returning the content page to the user, the content page program module 210 (shown in Figure 2) is appended to the content page. In one embodiment, in addition to the content page program module, a one line JavaScript script, pointing to Adsys.php module in the server 110, is added to the content page. The Adsys.php based configuration performs much the same functions as the frame page or Index.php method (in which, as described above, the initial page sent from the server 110 includes the frameset definition code), but uses a different method to install the frameset operating environment needed by the software. Appendix 10 shows the code in the Adsys.php module. In this embodiment, each web page in the site must be modified with the addition of a single line of JavaScript code at the top of the page text. The Adsys JavaScript line is added by the software through the administration screens. This single line accesses the Ad Server 110 to retrieve dynamically generated JavaScript code, which when executed at the browser, establishes the frameset, and loads the appropriate content into each frame just as the Index.php method does. The primary differences as compared to the Index.php method include (1) transparently installing the frameset environment such that the user cannot tell that the advertisement system is not in fact part of the original site content, (2) content caching is preferably not performed on the website content, and (3) click handling within the content frame is handled by JavaScript code that supersedes the normal browser and redirects clicks to the Advertisement Frame for handling.

[0065] In another embodiment, instead of adding this one line JavaScript script, each of the requested content pages are processed by the server 110 (in one

embodiment, the content processor 325 on the server 110) so that each URL in the content pages is rewritten. Each rewritten URL references a web page via the server 110, more specifically the adext.dll module of the server 110, rather than referencing the web page directly. In one embodiment, an advertisement ID is added to the content page.

[0066] Code in the frames page, e.g., JavaScript scripts, also sends a load advertisement request 413 to the advertisement program module 420. In response to this request, the advertisement frame causes an accessing 421 of the advertisement table 430 to get a list of advertisements associated with the content page being loaded into the content frame or most recently loaded into the content frame. In response to this request, the advertisement table 430 returns 431 the advertisement list to the advertisement program module 420. The advertisement program module 420 randomly selects an advertisement from the list. An advertisement page program module, e.g., the advertisement page program module 215 (shown in Figure 2), is appended to the selected advertisement and returned 422 to the client. In another embodiment, the advertisement ID in the content page is used to select an advertisement to be returned to the client.

[0067] The frames page also sends to the daemon program module 425 a request to load a blank page 414 into the daemon frame. In response, the daemon program module 425 returns 426 a blank static web page to the client. The static web page is blank in that it does not contain displayable or viewable content. The static web page includes scripts that survive transitions between displays of content in the content frame and advertisements in the advertisement frame. When a user clicks on the "HOLD" or "HOME" buttons described above, the advertisement page program module, e.g., the advertisement page program module 215 (shown in Figure 2), captures these click events and sends a request for script to be loaded into the daemon frame. Additionally, the advertisement page program module, e.g., the advertisement

page program module 215 (shown in Figure 2), also provides information regarding the click events to script in the daemon frame.

[0068] It is to be noted that the index program module 410, the content program module 415 and the advertisement program module 420 run on the server side, whereas the frames page program module 205, the content page program module 210, the advertisement page program module 215 run on the client side. It is also to be noted that, in one embodiment, the index program module 410 includes the frames processor 320 (shown in Figure 3). Similarly, in one embodiment, the content program module 415 includes the content processor 325 (shown in Figure 3). Similarly, in one embodiment, the advertisement program module 420 includes the advertisement processor 330 and advertisement hit processor 335 (shown in Figure 3).

[0069] Figure 5 is an interaction diagram for navigating a web site in conjunction with one embodiment of the method of the present invention. The interaction diagram of Figure 5 deals with navigating a web site enabled with the software of the present invention. More specifically, it deals with navigating such a web site after browsing the web site, such as, for example, as described in conjunction with Figure 4. When the user 505 clicks 506 on a hyperlink in the content frame, code, e.g., JavaScript scripts, in the content frame, notifies the advertisement frame to stop displaying the advertisement in the advertisement frame. Furthermore, the content program module returns 511 the web page associated with the hyperlink to the client. In one embodiment, the content page is processed and/or has scripts/information (such as scripts in the content page program module 210, the single line JavaScript script described above, or advertisement ID) appended thereto before it is returned to the client. This web page is loaded into the content frame, thus replacing the web page previously loaded in the content frame. Moreover, a start advertisement display request 512 is sent from the content program module 510 to the advertisement program module 515. In response to this request, the advertisement



program module 515 accesses 516 the advertisement table 525 to get a list of advertisements associated with the content being loaded into the content frame or most recently loaded into the content frame. In response to the request for the list of advertisements, the advertisement table returns 526 the advertisement list to the advertisement program module 515. The advertisement program module 515 randomly selects an advertisement from the list, and sends 517 the selected advertisement to the client. In one embodiment, scripts (such as those included in the advertisement program module 215) are appended to the advertisement before it is returned to the client. The new advertisement is filled into the advertisement frame and is shown immediately in the advertisement frame on the client screen.

[0070] It is to be noted that the content program module 510 and the advertisement program module 515, like their counterparts in Figure 4 (i.e., content program module 415 and advertisement program module 420, respectively) run on the server side. It is also to be noted that, in one embodiment, the content program module 510 includes the content processor 325 (shown in Figure 3). Similarly, in one embodiment, the advertisement program module 515 includes the advertisement processor 330 and advertisement hit processor 335 (shown in Figure 3).

[0071] The advertisement program module 515 also sends 518 a load logger script request to the daemon program module 520. Consequently, the daemon frame is loaded with the logger script. The daemon program module 520 sends 521 a request to the tracking table 530 to log the advertisement impression. As a result, the tracking table 530 records an impression event for the advertisement impression in the tracking table. Additionally, the daemon program module 520 returns 522 a blank page to the client.

[0072] Figure 6 is an interaction diagram for clicking on an advertisement in conjunction with one embodiment of the method of the present invention. The interaction diagram of Figure 6 deals with clicking on an advertisement after browsing a web site enabled with the software of the present invention. More

specifically, it deals with clicking on an advertisement after browsing the web site, such as, for example, as described in conjunction with Figure 4. When a user 605 clicks 606 on the Learn More (HOME) button, or the advertisement, in the advertisement frame, code, e.g., JavaScript scripts, in the advertisement program module 610, opens a new browser window 625 and loads 611 the destination URL for the advertisement into the new browser window 625. In response, the new browser window 625 returns 626 the advertiser's web page to the client. Moreover, code, e.g., JavaScript scripts, in the advertisement program module 610, sends a log click-through request 612 to the daemon program module 615 to record the click-through in the tracking table 620. In one embodiment, this request is sent at the same time as the request to load the advertisement URL. In response to the log click-through request 612, the daemon program module 615 sends an insert log record request 616 to the tracking table 620. As a result, the click-through log record is inserted into the tracking table 620. The tracking table 620 sends a return status message 621 to the daemon program module 615. The return status message indicates whether the click-through log record is successfully inserted into the tracking table. Furthermore, the daemon program module 615 returns 617 a blank web page to the client.

[0073] It is to be noted that the advertisement program module 610, like its counterpart in Figure 4 (i.e., advertisement program module 420) runs on the server side. It is also to be noted that, in one embodiment, the advertisement program module 610 includes the advertisement processor 330 and advertisement hit processor 335 (shown in Figure 3).

[0074] As noted in the configuration table above, when ExternalAds is set to Yes, then external advertisements (served by other advertisement servers, rather than the server 110) will be displayed. As also noted in the configuration table above, when ExternalAds is set to any other value (for example, No), then external advertisements will not be displayed on the local web site.

[0075] Figure 7 is an interaction diagram for exiting a web site in conjunction with one embodiment of the method of the present invention. The interaction diagram of Figure 7 deals with exiting a web site enabled with the software of the present invention. More specifically, it deals with exiting such a web site after browsing the web site, such as, for example, as described in conjunction with Figure 4. If ExternalAds is disabled (i.e., not set to Yes), then when the user 705 clicks 706 a hyperlink that refers to an external web site that is not enabled with the software of the present invention, code, e.g., JavaScript scripts, in the content program module 710, causes the external URL to be loaded 711 in the top level frames page, completely replacing the current web site content and advertisement with content from the external web site. In other words, as shown in Figure 7, clicking 706 the external hyperlink to a web site that is not enabled with the software of the present invention causes the content program module 710 to send a load external URL request 711 to the index program module 715. In response, the index program module 715 returns 716 the external web page to the client. If the ExternalAds is enabled (i.e., ExternalAds is set to Yes), then hyperlinks to an external web site are handled the same way as hyperlinks within the current web site which is enabled with the software of the present invention.

[0076] It is to be noted that the index program module 715 and the content program module 710, like their counterparts in Figure 4 (i.e., index program module 410 and content program module 415, respectively), run on the server side. It is also to be noted that, in one embodiment, the index program module 715 includes the frames processor 320 (shown in Figure 3). Similarly, in one embodiment, the content program module 710 includes the content processor 325 (shown in Figure 3).

[0077] Figure 8 shows a flowchart of major functions of the frames page builder script (index.php) in one embodiment of the present invention. Appendix 6 shows an example of JavaScript scripts included in the frames page builder script (index.php). As shown in Figure 8, when a user enters a site (e.g., a web site), in step

805, the server loads configuration into its memory. Thereafter in step 810, the server determines the target URL of the default page for the web site. In step 815, the server checks the license key to determine if the web site corresponding to the target URL is a licensed customer for using the software of the present invention. If it is determined in step 815 that the web site corresponding to the target URL is not a licensed customer, then the process continues at step 820. In step 820, the browser is directed to the target URL. Thereafter, the web page associated with the target URL is returned to the client. On the other hand, if it is determined in step 815 that the web site corresponding to the target URL is a licensed customer, then the process continues at step 825. In step 825, the software at the server 110 builds a frameset with frame content URL's including the target page, i.e., the web page originally requested by the user. In step 830, the server 110 sends the frameset and frames to the browser, more specifically, to the browser at the client.

[0078] In one embodiment, the content frame displays the original web site content, as modified by the Content Caching script (adload.php). In one embodiment, the Content Caching script is included in the content program module 415. Seamless integration with existing web site content is achieved by rewriting many HTML tags and attributes, and appending JavaScript code to manage the routing of hyperlink click events to the advertisement frame. In one embodiment, the frame loading script edits the content page to rewrite the HTML tags and append the JavaScript code to manage the routing of hyperlink click events to the advertisement frame. HTML tags may, for example, be rewritten so as to refer to the proper URL of the desired object. The appended JavaScript code is used to notify the advertisement frame of click events in the content page. In another embodiment, the above functionality is performed by Microsoft Visual Basic Script code or JScript code instead of JavaScript code. More generally, the above functionality need not be performed by a script language as it can be performed by other software code.

[0079] In one embodiment, in order to enhance performance, the Content Caching Script (adload.php) stores a fully rewritten version of each content page accessed in the "cache" folder. Given that the original page has not been modified and the page is being requested without query parameters, the cached version of the page is returned to the web browser, avoiding the overhead of reprocessing each web page to rewrite HTML tags and attributes each time it is accessed.

[0080] Figure 9 shows a flowchart of major functions of the content caching script (adload.php) in one embodiment of the present invention. Appendix 7 shows an example of JavaScript scripts included in the content caching script (adload.php). When the request to load the content frame is issued by the load content code in the frames page, the process proceeds to step 902. In step 902, the content caching script determines the server name for the web server with pages for the content frame. In step 904, the content caching script determines the target URL. In step 906, the content caching script determines whether a target server name is specified. If it is determined, in step 906, that a target server name is not specified, then the process proceeds to step 908. In step 908, the content caching script adds a target server name. Thereafter, the process proceeds to step 910. If it is determined in step 906 that a target server name is specified, then the process proceeds to step 910 without going through step 908. In step 910, the content caching script extracts the target server name. In step 912, the content caching script determines whether external advertisements are enabled in conjunction with the target URL. In other words, it determines whether advertisements are to be displayed when showing pages from the target URL. If external advertisements are enabled, then the process proceeds to step 914. In step 914, the content caching script reloads the index page with the target URL. If external advertisements are disabled or if the target URL is not external, then the process proceeds to step 916. In step 916, the content caching script removes query strings. In step 918, the content caching script removes anchor links. In step 920, the content caching script converts the target URL to a cache filename.

In step 922, the content caching script determines whether the target URL includes dynamic, rather than static, data. If, in step 922, it is determined that the target URL includes dynamic data, then the process proceeds to step 924. In step 924, the content caching script removes the requested page from cache. Thereafter, the process proceeds to step 926. If, in step 922, it is determined that the target URL does not include dynamic data, then the process proceeds to step 926 without going through step 924. In step 926, the content caching script determines if a cached file exists. If, in step 926, it is determined that a cached file exists, then the process proceeds to step 928. In step 928, the cached page is returned. If, in step 926, it is determined that a cached file is not found, then the process proceeds to step 930. In step 930, it is determined whether an original file is found. If, in step 930, it is determined that an original file is not found, then the process proceeds to step 932. In step 932, a message that the page is not found is displayed at the client. If, in step 930, it is determined that an original file is found, then the process proceeds to step 934. In step 934, the original HTML is loaded into memory. In step 936, the hyperlinks and image tags are rewritten. In step 938, JavaScript code snippets are inserted. These JavaScript code snippets include, for example, the scripts for handling mouse click events described above. In step 940, the processed HTML is written to cache. In step 942, the cached page is returned.

[0081] As noted above, in one embodiment, instead of rewriting the content page before sending it to the client, a one line JavaScript script is added to the content page. In such a case, the content page with the one line JavaScript script added thereto may be cached before it is sent to the client. Alternatively, it may be sent to the client without first being cached.

[0082] In one embodiment, the advertisement frame displays the advertisement for the most recently loaded web page using the Advertising Display Script (adview.php). Advertising content is retrieved from HTML files stored on the advertising server (internal or external), as rewritten by the Display Script to insure

that HTML Tags and Attributes for image, flash, video, etc. content refer to the correct URL within the web site. Additionally, JavaScript code (which in one embodiment is included in the advertisement page program module 215) is appended to manage timers and handle advertising click events. In one embodiment, a timer is used to control the amount of time during which an advertisement is displayed. For example, as noted above, in one embodiment, an advertisement is displayed until a duration timer runs out.

**[0083]** Figure 10 shows a flowchart of major functions of the advertisement display script flow chart (advview.php) in one embodiment of the present invention. Appendix 8 shows an example of JavaScript scripts included in the advertisement display script (advview.php). In step 1005, the advertisement display script determines the server name. In step 1010, it is determined whether a server is specified. If, in step 1010, it is determined that no server is specified, then the process proceeds to step 1015. In step 1015, a local server name is added. Thereafter, the process proceeds to step 1020. If, in step 1010, it is determined that a server is specified, then the process proceeds to step 1020 without going through step 1015. In step 1020, the target server name is extracted. In step 1025, query strings are removed. In step 1030, anchor links are removed. In step 1035, a list of matching web sites are read. In step 1040, a list of advertisements for the matching web sites are read. In step 1045, an advertisement is randomly selected from the list of advertisements. In step 1050, the advertisement content is loaded into browser memory. In step 1055, the hyperlink and image tags are rewritten. In step 1060, an interstitial or side bar HTML header is inserted. In step 1065, JavaScript code snippets are inserted. In step 1070, the updated content is returned to the browser.

**[0084]** As noted above, in one embodiment, a advertisement ID is added to the content page before it is sent to the client 105. In such a case, the advertisement is selected, not randomly, but based on the advertisement ID.

[0085] It is to be noted that the present invention allows displaying advertisements in the split screen or interstitial mode without hard recoding of the web pages in conjunction with which the advertisements are delivered. In other words, the present invention can be practiced without manual script inserting or re-programming of web pages to enable different modes of advertisement display. The present invention is completely soft coded and can be modified or removed through administration screens.

[0086] While the present invention has been described in the context of content and advertisement delivery, it is not limited to use in this context. For example, the method of the present invention can be applied to an e-mail system, a file transfer, a bulletin board, a chat room, or a closed circuit television. Additionally, while the present invention has been described with respect to one advertisement, one skilled in the art would recognize that multiple advertisement frames could be included in the frames page. In such a case, it may be advantageous to include additional script program code to facilitate the management of multiple advertisement pages. Such additional code may, for example, coordinate display of multiple advertisements, either sequentially or simultaneously, in the time period between the display of content pages.

[0087] While the present invention has been particularly described with respect to the illustrated embodiments, it will be appreciated that various alterations, modifications and adaptations may be made based on the present disclosure, and are intended to be within the scope of the present invention. While the invention has been described in connection with what are presently considered to be the most practical and preferred embodiments, it is to be understood that the present invention is not limited to the disclosed embodiment but, on the contrary, is intended to cover various modifications and equivalent arrangements included within the scope of the appended claims.



## APPENDIX 1      JAVASCRIPT scripts in a typical HOLDER PAGE

```

<script LANGUAGE='jscript'>
<!--
fl="%s";
var IsMultiAd=%d;
var IsShowOnLoad=%d;
var waitInterval=%d*1000;
var ExternalSiteAdLoad=%d;
var LockAd=false;
var LoadAdAllowed=false;
var IsBannerAd=%s;
var IsLimitedUnload=false;
var IsHorzAd=%s;
var AdP=%d;
var ContPercent=%d;
var AdPercent=%d;
var IsMultiAdInNewWindow=%s;
var IsMultiAdInNewWindowLoaded=false;
var IsContentLoaded=false;
var ContentURL;
var IsAdFromBannerInContent=true;
var KeepAd=false;
//-->
</script>
<script LANGUAGE="jscript">
<!--
var flag=3;
var IsBegin=true;
var AdID="%d";
var AdID2="%d";
var AdID3="%d";
var AdID4="%d";
var AdIDOld=0;
function haction(hrefParam,IsExternal) {
    IsExternalSite=IsExternal;
    if (top.ExternalSiteAdLoad)
        LockAd=!IsExternal;
    LoadAdAllowed=true;
    if(!IsExternal)
        window.top.top1.location.href = hrefParam;
    else
        window.location.href = hrefParam;
}
function LoadEmptyFirstImplem(PageURL)
{
    window.top.top1.document.body.innerHTML='';
    window.top.top1.document.bgColor='white';
    window.top.top1.location.href=PageURL;
}

function LoadEmptyFirst(PageURL)
{
    window.setTimeout("top.LoadEmptyFirstImplem(\""+PageURL+"\"),100);
}

```

```

function window_onclick()
{
    top.inclick=true;
}
function StopAllFlashOb()
{
    StopFlashOb(top.bottom1);
    StopFlashOb(top.bottom2);
    StopFlashOb(top.bottom3);
    StopFlashOb(top.bottom4);
}
function StopFlashOb(windowob)
{
    try
    {
        windowob.flashob.Stop();
        windowob.flashob.FrameNum=20;
    }
    catch(e)
    {
    }
    try
    {
        windowob.RemoveMovie();
    }
    catch(e)
    {
    }
}
function SetFramesOriginal()
{
    if(KeepAd)
        return;
    ob=top.document.body;
    str=ContPercent+","+AdPercent+",0,0,0";
    if (IsHorzAd)
        ob.rows=str;
    else
        ob.cols=str;
    if (!top.IsBannerAd)
    {
        window.setTimeout("StopAllFlashOb()",1);
    }
}
function LoadAdPage(AdFrame,PageID)
{
    if (!PageID)
        return;
    if (AdFrame==1)
    {
        //      if (!IsAdFromBannerInContent)

        window.top.bottom1.location.href="/dpbin/AdExt.dll?Ad&"+PageID+"&0";
        //      else

```

```

//
    window.top.bottom1.location.href="/dpbin/AdExt.dll?BannerAdFromURL&"+top.
ContentURL;
    }
    else if (AdFrame==2)

        window.top.bottom2.location.href="/dpbin/AdExt.dll?Ad&"+PageID+"&0";
    else if (AdFrame==3)

        window.top.bottom3.location.href="/dpbin/AdExt.dll?Ad&"+PageID+"&0";
    else if (AdFrame==4)

        window.top.bottom4.location.href="/dpbin/AdExt.dll?Ad&"+PageID+"&0";
}
function LoadAdPagesWhenContentIsLoaded()
{
    if (KeepAd==true || top.IsShowOnLoad==false &&
top.IsContentLoaded==false)
    {
        window.setTimeout("LoadAdPagesWhenContentIsLoaded()",100);
        return;
    }
    window.top.bottom1.document.body.innerHTML='';
    window.top.bottom1.document.bgColor='white';
    try
    {
        window.top.bottom2.document.body.innerHTML='';

        window.top.bottom2.document.bgColor='white';
        window.top.bottom3.document.body.innerHTML='';
        window.top.bottom3.document.bgColor='white';
        window.top.bottom4.document.body.innerHTML='';
        window.top.bottom4.document.bgColor='white';
    }
    catch(e)
    {
        window.setTimeout("LoadAdPage(1,"+top.AdID+")",500);
%s        window.setTimeout("LoadAdPage(2,"+top.AdID2+")",1000);
%s        window.setTimeout("LoadAdPage(3,"+top.AdID3+")",1500);
%s        window.setTimeout("LoadAdPage(4,"+top.AdID4+")",2000);
    }

function LoadChildCont(IsAd)
{
    if (top.IsMultiAdInNewWindow)
    {
        top.mywin =
window.open("/dpbin/AdExt.dll?MultAdInNewWindow","win",'left=4000,top=4000,titl
ebar=1,toolbar=1,location=1,directories=1,status=1,menubar=1,scrollbars=1,resiz
able=1,width=400,height=300',1);
        top.mywin.blur();
        top.mywin.moveTo(window.screenLeft,window.screenTop);
        return;
    }
}

```

```

        if(((top.IsBegin && top.ExternalSiteAdLoad) || (!top.ExternalSiteAdLoad))
        && (IsAd || top.IsShowOnLoad))
        {
            SetFramesOriginal();
            top.IsBegin=false
            top.LoadAdPagesWhenContentIsLoaded();
//
// window.top.bottom1.location.href="/dpbin/AdExt.dll?Ad&"+top.AdID+"&0";
// window.top.bottom2.location.href="/dpbin/AdExt.dll?Ad&"+top.AdID2+"&0";
// window.top.bottom3.location.href="/dpbin/AdExt.dll?Ad&"+top.AdID3+"&0";
// window.top.bottom4.location.href="/dpbin/AdExt.dll?Ad&"+top.AdID4+"&0";
//
        }

function LoadFirstContent()
{
    FrameName=f1;
    top.IsBegin=true
    window.top.top1.location.href = "/dpbin/AdExt.dll?D&"+FrameName+"&0&";
    top.AdID="%d";
    top.AdID2="%d";
    top.AdID3="%d";
    top.AdID4="%d";
}

function LoadChild(IsAd)
{
    if (top.IsBegin!=true)
        timeoutvar=waitInterval;
    else
        timeoutvar=0;
    if(!top.IsShowOnLoad || LockAd)
        timeoutvar=0;
    if ((top.IsBegin==true || top.IsShowOnLoad))
        window.setTimeout("LoadChildCont("+IsAd+")",timeoutvar);
}

function ActivateAd(AdNumber)
{
    var ob;
    framesob = new Array(5)
    if(AdP)
        framesob[0]=ContPercent;
    else
        framesob[0]=0;
    for (i=1;i<5;i++)
    {
        framesob[i]=0;
    }
    if(AdP)
        framesob[AdNumber]=AdPercent;
    else
        framesob[AdNumber]=10;
    var obj;

```

```

    if (AdNumber==1)
        obj=top.bottom1;
    if (AdNumber==2)
        obj=top.bottom2;
    if (AdNumber==3)
        obj=top.bottom3;
    if (AdNumber==4)
        obj=top.bottom4;
    try
    {
        obj.flashob.Rewind();
        obj.flashob.Play();
    }
    catch(e)
    {
    }
    try
    {
        obj.PlayMovie();
    }
    catch(e)
    {
    }
}

str="";
for (i=0;i<5;i++)
{
    str=str+framesob[i];
    if (i<4)
        str+=", ";
}
if (IsHorzAd)
    ob=top.document.all.tags("frameset").item(0).rows=str;
else
    ob=top.document.all.tags("frameset").item(0).cols=str;
/*
if (AdNumber==1)
    ob="0,10,0,0,0";
else if (AdNumber==2)
    ob="0,0,10,0,0";
else if (AdNumber==3)
    ob="0,0,0,10,0";
else if (AdNumber==4)
    ob="0,0,0,0,10";*/
/*
try
{
    bottom1.flashob.Play();
}
catch(e)
{
}*/
if (!AdP)
    window.setTimeout("SetFramesOriginal()",waitInterval);
}
function LoadAd(IsAd)
{
    if (!LoadAdAllowed)
        return;

```

```

if(IsAd || top.IsShowOnLoad)
{
    var frm=top.document.frames;
    var pl=frm(1);
    if (!LockAd)
    {
        if (!AdP && !top.IsMultiAdInNewWindow)
        {
            try
            {
                top.bottom1.flashob.Rewind();
                top.bottom1.flashob.Play();
            }
            catch(e)
            {
            }
            try
            {
                top.bottom1.PlayMovie();
            }
            catch(e)
            {
            }

            if (IsHorzAd)

top.document.all.tags("frameset").item(0).rows="0,10,0,0,0";
            else

top.document.all.tags("frameset").item(0).cols="0,10,0,0,0";
        }
        if(top.ExternalSiteAdLoad && top.IsMultiAdInNewWindow)
        {
            try
            {
                top.mywin.waitInterval=waitInterval;
                window.setTimeout("top.mywin.RotateAd()",1);
            }
            catch(e)
            {
            }

            return;
        }
        if(top.ExternalSiteAdLoad)
        {
            //mywin =
window.open("AdExt.dll?Ad&" + top.AdIDOld, "win", 'width=1,height=1,left=4000,top=4
000,titlebar=0,toolbar=0,location=0,directories=0,status=0,menubar=0,scrollbars
=1,resizable=1',1);

            window.setTimeout("some()",1);
        }
        open_window("AdExt.dll?Ad&" + top.AdIDOld)
        window.setTimeout("delay(waitInterval)",0);
        alert('oo')
    }
}

```

```

    }
}
timeoutvar=waitInterval;
if(LockAd)
    timeoutvar=0;
LoadAdAllowed=false;
if(IsAd && (!top.IsShowOnLoad) && (!IsMultiAdInNewWindow))
{
    window.setTimeout("LoadChildCont("+IsAd+")",timeoutvar);
}
}

function open_window(url) {
mywin =
window.open(url,"win",'titlebar=0,toolbar=0,location=0,directories=0,status=0,m
enubar=0,scrollbars=1,resizable=1,width=546,height=500',1);
}
//-->
</script>
<script LANGUAGE="jscript">
<!--
var mywin;
function delay(del)
{
    var d = new Date();
    var ms = d.getTime();

    a=0;
    for (i=0;i<1000000000;i++)
    {
        var d1 = new Date();
        if (d1.getTime()>ms+del)
            break;
        window.setTimeout("some()",1000);
    }
}
function some()
{
    // mywin =
window.open("/dpbin/AdExt.dll?Ad"+top.AdIDOld,"win",'width=1,height=1,left=400
0,top=4000,titlebar=1,toolbar=1,location=1,directories=1,status=1,menubar=1,scr
ollbars=1,resizable=1',1);
    // mywin =
window.open("/dpbin/AdExt.dll?Ad"+top.AdIDOld,"win",'left=0,top=0,titlebar=1,t
oolbar=1,location=1,directories=1,status=1,menubar=1,scrollbars=1,resizable=1',
1);
    top1.mywin =
window.open("/dpbin/AdExt.dll?Ad"+top.AdIDOld+"&1","win",'left=4000,top=4000,t
itlebar=1,toolbar=1,location=1,directories=1,status=1,menubar=1,scrollbars=1,re
sizable=1',1);
    top1.mywin.blur();
    top1.IsLimitedUnload=true;
}

```

```

        top1.waitInterval=waitInterval;
//      mywin.flashob.Stop();
//      window.focus();
//      mywin.resizeTo(546, 500);
//      mywin.moveBy(-2, -30)

//      mywin.resizeTo(mywin.screen.availWidth, mywin.screen.availHeight);
//      mywin.moveTo(0, 0);

//      mywin.Sp();
try
{
//      top1.mywin.flashob.Play();
/*      mywin.flashob.Stop();
      mywin.flashob.FrameNum=10;
      mywin.blur();
      mywin.MyFunction(window); */
}
catch(e)
{
}
}

function baction(hrefParam, IsExternal) {
    if (top.ExternalSiteAdLoad)
        LoadAdAllowed=true;
    if (!IsExternal)
        window.location.href = hrefParam;
    else
        window.top.top1.location.href = hrefParam;
}

function RefreshChildCont()
{
    if (((top.IsBegin && top.ExternalSiteAdLoad) || (!top.ExternalSiteAdLoad))
    && (IsAd || top.IsShowOnLoad))
    {
        top.document.body.cols="0,10"
        top.IsBegin=true
        window.top.bottom1.location.href="/dpbin/AdExt.dll?D&"+top.AdID;
    }
}

function LoadSecondContent()
{
    FrameName=f1;
    top.IsBegin=true
    window.top.top1.location.href = "/dpbin/AdExt.dll?D&"+FrameName+"&0&";
    top.AdID="3";
}

function PlayAd(IsAd)
{
    if (top.IsBegin==true)

```



```

        timeoutvar=waitInterval;
    else
        timeoutvar=0;
    if(top.IsShowOnLoad && LockAd)
        timeoutvar=0;
    if ((top.IsBegin==true || top.IsShowOnLoad))
        window.setTimeout("RefreshChildCont("+IsAd+")",timeoutvar);
}

```

```

function ShutAd(IsAd)
{
    if (LoadAdAllowed)
        PlayAd(IsAd);

    if(IsAd && top.IsShowOnLoad)
    {
        var frm=top.document.frames;
        var pl=frm(1);
        if (!LockAd)
            top.document.all.tags("frameset").item(0).cols="0,10";
    }
    timeoutvar=waitInterval;
    if(!LockAd)
        LoadAdAllowed=false;
    if(IsAd && (!top.IsShowOnLoad))
    {
        window.setTimeout("PlayAd("+IsAd+")",timeoutvar);
    }
}

```

```

//-->
</script>
<html>
<head>
<title>holder</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>

<SCRIPT language=JavaScript>
<!--
str=ContPercent+", "+AdPercent+", 0,0,0";
if (IsHorzAd)
    document.write('<frameset rows='+str+' frameborder="0"
framespacing="0">');
else
    document.write('<frameset cols='+str+' frameborder="0"
framespacing="0">');
//-->

</SCRIPT>

<frame name="top1" src="/dualproject/frame1.htm&xxxxx=xxxxx&">
<frame name="bottom1" src="/dualproject/frame2.htm&xxxxx=xxxxx&">

```

```
%s<frame name="bottom2" src="/dualproject/frame2.htm&xxxxxx=xxxxxx&">
  <frame name="bottom3" src="/dualproject/frame2.htm&xxxxxx=xxxxxx&">
  <frame name="bottom4" src="/dualproject/frame2.htm&xxxxxx=xxxxxx&">
%s
  <noframes>
  <body>
  <p>This page uses frames, but your browser doesn't support them.</p>
  </body>
  </noframes>
</frameset>
</html>
```

## APPENDIX 2      JAVASCRIPT scripts in a typical CONTENT PAGE

```
<script ID="clientEventHandlersJS" LANGUAGE="jscript">
<!--
top.inclick=false;
var mywin;
var IsLimitedUnload;
var waitInterval;
```

```
function document_onmouseenter() {
    return mouseaction() ;
}
function document_onmouseleave()
{
    return mouseaction() ;
}
function document_onmouseover()
{
    return mouseaction() ;
}
function document_onmouseout()
{
    return mouseaction() ;
}
function document_onmousemove()
{
    return mouseaction() ;
}
function mouseaction()
{
    window.status="";
    window.event.cancelBubble=false;
    window.defaultStatus="";
    window.event.returnValue=true;
    return false;
    return true;
}
```

```
function window_onbeforeunload()
```

```

{
    if (!top.IsShowOnLoad)
        top.IsContentLoaded=false;
    if (top.IsMultiAd)
    {
        //top.LoadChildCont(true)
        return;
    }
    top.AdIDOld=top.AdID;
    top.AdID="%d";
    top.AdID2="%d";
    top.AdID3="%d";
    top.AdID4="%d";
    top.ContentURL="%s";

    IsAd=%d;
    if (top.AdID=="0")
        top.LockAd=true;
    if (top.inclick==false || top.ExternalSiteAdLoad==0)
    {
        top.LoadAdAllowed=true;
        top.LockAd=false;
        if (top.AdID=="0")
            top.LockAd=true;
        // alert('pp');
    }
    // alert('pp1');

    top.LoadAd(IsAd);
}

function window_onload() {
window.document.body.onmouseover=document_onmouseover;
window.document.body.onmouseout=document_onmouseout;
window.document.body.onmousemove=document_onmousemove;
window.document.body.onmouseenter=document_onmousemove;
window.document.body.onmouseleave=document_onmouseleave;
/* if (top.IsShowOnLoad)
{
    top.AdID="%d";
}
else if (top.IsBegin==true)*/
top.AdID="%d";
top.ContentURL="%s";

if (!top.IsShowOnLoad)
    top.IsContentLoaded=true;

IsAd=%d;
if (top.IsMultiAdInNewWindow==true &&
top.IsMultiAdInNewWindowLoaded==false)
{
    top.IsMultiAdInNewWindowLoaded=true;
    window.setTimeout("top.LoadChildCont(true)", 0);
    return;
}
if (top.IsMultiAdInNewWindow==true )

```

```
        return;
    if (top.IsMultiAd)
    {
        top.AdID="%d";
        top.AdID2="%d";
        top.AdID3="%d";
        top.AdID4="%d";
        window.setTimeout("top.LoadChildCont(true)",0);
    //    top.LoadChildCont(true)
    }

    if ((!top.IsMultiAd) && (!top.IsMultiAdInNewWindow) && (top.IsBegin==true ||
    top.IsShowOnLoad))
        top.LoadChild(IsAd);
    }

    function window_onunload()
    {
        if (!IsLimitedUnload)
            return;

        window.focus();
        mywin.resizeTo(mywin.screen.availWidth,mywin.screen.availHeight);
        mywin.moveTo(0,0);
        mywin.focus();
        try
        {
            mywin.flashob.Play();
        }
        catch(e)
        {
        }
    }
    //    mywin.TimeoutCloseWin(waitInterval);
    mywin.setTimeout("CloseMyWin()",waitInterval);
}
//-->
</script>
```

## APPENDIX 3      JAVASCRIPT Scripts in a typical AD PAGE

```

<script ID="clientEventHandlersJS" LANGUAGE="jscript">
<!--
var IsSelf=%d;
function Hold_onclick() {
    if (IsSelf==1)
    {
        try
        {
            flashob.Play();
        }
        catch(e)
        {
        }
    }
    else
    {
        if (Hold.innerText=="Hold-Ad")
        {
            Hold.innerHTML="<a href=\"javascript:Hold_onclick()\"><font
size=\"1\">Resume-Ad</a>";
            top.KeepAd=true;

            emptyframe.location.href="%s/dpbin/AdExt.dll?AdHit&HitType=2&AdID=%d&Page
=s";
        }
        else
        {
            Hold.innerHTML="<a href=\"javascript:Hold_onclick()\"><font
size=\"1\">Hold-Ad</a>";
            top.KeepAd=false;
            top.SetFramesOriginal();
        }
    }
}

function HomePage_onclick() {
open_window ("%s/dpbin/AdExt.dll?AdHit&HitType=1&AdID=%d&Page=s");
}
function open_window(url) {
//mywin =
window.open(url,"win",'titlebar=0,toolbar=0,location=0,directories=0,status=0,m
enubar=0,scrollbars=1,resizable=1,width=546,height=500',1);
}

```

```
mywin =
window.open(url,"win",'titlebar=1,toolbar=1,location=1,directories=1,status=1,m
enubar=1,scrollbars=1,resizable=1,width=546,height=500',1);
}
var waitInterval=%d*1000;
function TryToStopAgain()
{
    flashob.FrameNum=10;
    flashob.Stop();
}

function window_onload()
{
    if (top.IsBannerAd)
    {
        try
        {
            top.bottom1.flashob.Play();
        }
        catch(e)
        {
        }
        try
        {
            top.bottom1.PlayMovie();
        }
        catch(e)
        {
        }
    }

    try
    {
        if (top.IsBannerAd)
        {
            flashob.Play();
        }
        else
        {
            flashob.Stop();
            flashob.FrameNum=10;
            flashob.Stop();
            window.setTimeout("TryToStopAgain()",100);
        }
    }
    catch(e)
    {
    }
    try
    {
        if (top.IsBannerAd)
        {
            PlayMovie();
        }
    }
```

```
        else
        {
            RemoveMovie();
        }
    }
    catch(e)
    {
    }
}

/*  if (IsSelf==0)
{
    try
    {
        flashob.Stop();
    }
    catch(e)
    {
    }
}
else if (IsSelf==1)
{
    try
    {
        flashob.Play();
    }
    catch(e)
    {
    }
} */

//  if (IsSelf==1)
//      window.setTimeout("CloseMyWin()",waitInterval);
}
function TimeoutCloseWin(waitInterval)
{
    //  window.setTimeout("CloseMyWin()",waitInterval);
}
function CloseMyWin()
{
    if (IsSelf==1)
    {
        window.blur();
    }
    try
    {
        window.flashob.FrameNum=10;
        window.flashob.Stop();
    }
    catch(e)
    {
    }
}
}
```



```
//-->
</script>
```

#### Appendix 4      Logger.php

```
<%
include((isset($_SERVER[DOCUMENT_ROOT])?$_SERVER[DOCUMENT_ROOT]:dirname(dirname(
dirname($_SERVER[PATH_TRANSLATED])))) . "/sofcast/tools/config.php"); %>
<%

$agent = $_HTTP_USER_AGENT;
$isie = !(strpos($agent, "MSIE") === false);
$ismoz4 = !(strpos($agent, "Mozilla/4") === false);
$ismoz5 = !(strpos($agent, "Mozilla/5") === false);
$ismoz = $ismoz4 || $ismoz5;

$extlist = array(
    "pl",
    "cgi",
    "php",
    "jsp",
    "asp",
    "htm",
    "html",
    "shtm",
    "shtml"
);

function scandir($path) {
    global $extlist;
    global $DOCUMENT_ROOT;
    $dir = opendir($path);
    while ($fn = readdir($dir)) {
        if ($fn == ".") continue;
        if ($fn == "..") continue;
        $rfn = "$path$fn";
        if (is_dir("$rfn")) {
            scandir("$rfn/");
        } else {
            $ext = "";
            if ($pos = strrpos($fn, ".")) {
                $ext = substr($fn, $pos+1);
            }
            if (in_array($ext, $extlist)) {
                $tmp = substr($rfn, strlen($DOCUMENT_ROOT));
                if (substr($tmp, 0, 7) == "/sofcast/tools/") continue;
                if (substr($tmp, 0, 5) == "/sofcast/bin/") continue;
                echo "$tmp\n";
            }
        }
    }
}
```

```
}

if (isset($scan)) {
    scandir($DOCUMENT_ROOT. "/");
    return;
}

if (isset($aucheck)) {
    $fp = fopen("$DOCUMENT_ROOT$aucheck", "r");
    $dat = fread($fp, 65535);
    fclose($fp);
    if (strstr($dat, "/sofcast/bin/adunder.php")) {
        %> under<%
    }
    if (strstr($dat, "/sofcast/bin/adafter.php")) {
        %> under<%
    }
    if (strstr($dat, "/sofcast/bin/adshow.php")) {
        %> box<%
    }
    if (strstr($dat, "/sofcast/bin/adsys.php")) {
        %> sys<%
    }
    return;
}

if (isset($save)) {
    $test = md5("path=$save\ndata=$data");
    $data = stripslashes($data);
    if ($hash != $test) {
        $test = md5("path=$save\ndata=$data");
        if ($hash != $test) {
            echo "key mismatch !!<br>";
            return;
        }
    }
    $size = strlen($data);
    echo "Key = $key<br>";
    echo "File = $save<br>";
    echo "Data = $size<br>";
    $fp = fopen("$DOCUMENT_ROOT$save", "w");
    fwrite($fp, $data);
    fclose($fp);
    return;
}

if (isset($file)) {
    $fp = fopen("$DOCUMENT_ROOT$file", "r");
    $dat = fread($fp, 65535);
    fclose($fp);
    echo $dat;
    return;
}

include("$DOCUMENT_ROOT/sofcast/bin/siteinfo.php");
include("$DOCUMENT_ROOT/sofcast/bin/logmod.php");
```

```

if (isset($dat)) {
    $sql = "select p1.sid, p1.gid, p4.gname, p1.acid, p2.cid, p5.cname,
p2.acname, p2.prcimp, p2.prcclk, p2.prchld, p2.prcmrk, p3.siteurl,
sum(p1.cntimp), sum(p1.cntclk), sum(p1.cnthld), sum(p1.cntmrk) from tracking as
p1, campaigns as p2, sites as p3, pgroups as p4, advertisers as p5 where
p5.cid=p2.cid and p4.gid=p1.gid and p3.sid=p1.sid and p1.acid=p2.acid and
tick='$dat' group by p1.acid, p1.gid order by p2.cid, p1.acid";
    $res = mysql_query($sql, $connection);
    while
    (list($sid,$gid,$gname,$acid,$cid,$cname,$acname,$pimp,$pclk,$phld,$pmrk,$surl,
$scimp,$cclk,$schld,$cmrk) = mysql_fetch_row($res)) {
        echo
        "'$surl','$gname','$cname','$acname','$pimp','$pclk','$phld','$pmrk','$scimp','$cclk','$schld',
$cmrk\n";
    }
    return;
}

if (isset($grp)) {
    // handle JS-less box ads
    if (!isset($url)) $url = $_SERVER[HTTP_REFERER];
    if ($url == "") $url = "http://$snm";
    list($new,$sid,$enablehold,$frameside) = siteinfo($url);
    $act = "clk";
    if (isset($ord)) {
        $sql = "select cid, aid, gid from boxclick where sid=$sid and
ipa='$REMOTE_ADDR' and ord='$ord'";
        $res = mysql_query($sql, $connection);
        list($acid, $aid, $gid) = mysql_fetch_row($res);
        $sql = "delete from boxclick where shown < (now() - interval 1
day)";
        $res = mysql_query($sql, $connection);
    } else {
        $sql = "select gid from pgroups where sid=$sid and gname='$grp'";
        $res = mysql_query($sql, $connection);
        list($gid) = mysql_fetch_row($res);
        $sql = "select cid, aid from boxclick where sid=$sid and
ipa='$REMOTE_ADDR' and gid=$gid";
        $res = mysql_query($sql, $connection);
        list($acid, $aid) = mysql_fetch_row($res);
    }
    $sql = "select alink from adverts where aid=$aid";
    $res = mysql_query($sql, $connection);
    list($lnk) = mysql_fetch_row($res);
    // fall through to normal click processing
}
header("X-Sofcast-SID: $sid");
header("X-Sofcast-GID: $gid");
header("X-Sofcast-CID: $acid");
header("X-Sofcast-AID: $aid");

if ($lnk == "") {
    $lnk = logevent($act, $sid, $gid, $acid, $aid);
} else {
    logevent($act, $sid, $gid, $acid, $aid);
}
header("X-Sofcast-LNK: $lnk");

```

[illegible]

APPENDIX 5 ADEXT.DLL ----- SOURCE FILES

---

Appendix 5a adext.clw

; CLW file contains information for the MFC ClassWizard

```
[General Info]
Version=1
LastClass=CAAdExtExtension
LastTemplate=CDialog
NewFileInclude1=#include "stdafx.h"
NewFileInclude2=#include "AdExt.h"
```

```
ClassCount=1
Class1=CAAdExtExtension
ResourceCount=0
LastPage=0
```

```
[CLS:CAAdExtExtension]
Type=0
HeaderFile=ADEXT.h
ImplementationFile=ADEXT.cpp
Filter=N
LastObject=CAAdExtExtension
```

## APPENDIX 5b Adext.cpp

```
// ADEXT.CPP - Implementation file for your Internet Server
// AdExt Extension
```

```
#include "stdafx.h"
#include "AdExt.h"
#include "..\..\include\coll.h"
#include "..\..\include\htmlx.h"
```

```
#import "..\DFComServer2\DFComServer2.tlb" no_namespace
#import "..\..\htmlbld\htmlbld.tlb" no_namespace
#import "C:\Program Files\Common Files\System\ado\MSADO20.tlb" \
    no_namespace rename("EOF", "EndOfFile")
```

```
////////////////////////////////////
// The one and only CWinApp object
// NOTE: You may remove this object if you alter your project to no
// longer use MFC in a DLL.
```

```
CWinApp theApp;
```

```
////////////////////////////////////
// command-parsing map
```

```
BEGIN_PARSE_MAP(CAdExtExtension, CHttpServer)
    // TODO: insert your ON_PARSE_COMMAND() and
    // ON_PARSE_COMMAND_PARAMS() here to hook up your commands.
    // For example:

    ON_PARSE_COMMAND(Default, CAdExtExtension, ITS_EMPTY)
    DEFAULT_PARSE_COMMAND(Default, CAdExtExtension)
    ON_PARSE_COMMAND(Holder, CAdExtExtension, ITS_PSTR ITS_PSTR)
    ON_PARSE_COMMAND_PARAMS("target== fvfvf==")

    ON_PARSE_COMMAND(D, CAdExtExtension, ITS_PSTR ITS_I2 ITS_PSTR)
    ON_PARSE_COMMAND_PARAMS("target== AdNum=0 etc==")

    ON_PARSE_COMMAND(Ad, CAdExtExtension, ITS_I4)
```

```

ON_PARSE_COMMAND_PARAMS("AdID=0 ")

ON_PARSE_COMMAND(AdHit, CAdExtExtension, ITS_I2 ITS_I4 ITS_PSTR)
ON_PARSE_COMMAND_PARAMS("HitType=1 AdID=0 Page=~")

END_PARSE_MAP(CAdExtExtension)

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// The one and only CAdExtExtension object

CAdExtExtension theExtension;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CAdExtExtension implementation

CAdExtExtension::CAdExtExtension()
{
    m_nAdPercent=0;
}

CAdExtExtension::~CAdExtExtension()
{
}

void CAdExtExtension::RedirectURL(CHttpServerContext *pCtxt, LPCTSTR url)
{
    DWORD size = _tcslen(url);

    pCtxt->m_bSendHeaders = FALSE;
    pCtxt->m_pECB->ServerSupportFunction(pCtxt->m_pECB->ConnID,
                                        HSE_REQ_SEND_URL_REDIRECT_RESP,
                                        (void *)url,
                                        &size,
                                        NULL);
}

BOOL CAdExtExtension::GetExtensionVersion(HSE_VERSION_INFO* pVer)
{
    // Call default implementation for initialization
    CHttpServer::GetExtensionVersion(pVer);

    // Load description string
    TCHAR sz[HSE_MAX_EXT_DLL_NAME_LEN+1];
    ISAPIVERIFY(::LoadString(AfxGetResourceHandle(),
                            IDS_SERVER, sz, HSE_MAX_EXT_DLL_NAME_LEN));
    _tcscpy(pVer->lpszExtensionDesc, sz);
    return TRUE;
}

BOOL CAdExtExtension::TerminateExtension(DWORD dwFlags)
{
    // extension is being terminated
    //TODO: Clean up any per-instance resources

```



```

        return TRUE;
    }

    //////////////////////////////////////
    // CAdExtExtension command handlers

void CAdExtExtension::Default(CHttpServerContext* pCtxt)
{
    StartContent(pCtxt);
    WriteTitle(pCtxt);

    *pCtxt << _T("This default message was produced by the Internet");
    *pCtxt << _T(" Server DLL Wizard. Edit your CAdExtExtension::Default()");
    *pCtxt << _T(" implementation to change it.\r\n");
    EndContent(pCtxt);
}

CString CAdExtExtension::LoadHTML(UINT id)
{
    LPCTSTR lpszResourceName = MAKEINTRESOURCE(id);
    HINSTANCE hInst = AfxFindResourceHandle(lpszResourceName, RT_DLGINIT);
    HRSRC h = ::FindResource(hInst, lpszResourceName, RT_HTML);
    HGLOBAL hResource = NULL;
    void* lpResource = NULL;
    CString strOut;
    if (h != NULL)
    {
        // load it
        hResource = LoadResource(hInst, h);
        if (hResource == NULL)
        {
            TRACE0("HTML resource was found, but could not be
loaded.\n");
            return strOut;
        }

        // lock it
        lpResource = LockResource(hResource);
    }
    strOut=(LPCTSTR)lpResource;
    if (lpResource != NULL && hResource != NULL)
    {
        UnlockResource(hResource);
        FreeResource(hResource);
    }
    return strOut;
}

void CAdExtExtension::Holder(CHttpServerContext * pCtxt, LPCTSTR target,
LPCTSTR fvfvf)
{
    /* IHtmlBuilderObPtr HTMLob(__uuidof(HtmlBuilderOb));
    CString filestr(target);
    filestr+="&XXXXXX=XXXXX&";
    HTMLob->PutURL(filestr.AllocSysString());
    HTMLob->Load();*/

    long lAdID=0;

```

```

    long lAdIDFirst=0;
    CString strIsBannerAd("false");
    if (m_nAdPercent!=100)
        strIsBannerAd="true";
    CString strServer;
    int ShowAd=1;
    IAdapterManager2Ptr AdMgr(__uuidof(AdManager2));
    int IsLimitedAd=AdMgr->IsLimitedAd;
    m_nAdPercent=AdMgr->GetBannerAdPercent();
    int ContentShownOnLoad=0;
    long lSleepInt=100;
    // if (!IsLimitedAd)
    {
        GetDinfo(pCtxt, 0, target, lAdID, lAdIDFirst, ShowAd, strServer);
        ContentShownOnLoad=AdMgr->GetAdOnContentLoad();
        lSleepInt=AdMgr->GetAdSleepInterval();
    }
    // else
    //     lSleepInt=AdMgr->LimitedAdDelay;

    //     AdMgr->BuildAds();
    #if _NODB
        lSleepInt=2;
    #endif
    CString strHolderHTMLfmt=LoadHTML(IDR_HTML_HOLDER);
    CString strHolderHTML;
    int AdP, ContP;

    if (m_nAdPercent==0)
    {
        AdP=0;
        ContP=10;
    }
    else
    {
        AdP=100;
        ContP=(10000/m_nAdPercent-100);
    }
    strHolderHTML.Format(strHolderHTMLfmt, target, ContentShownOnLoad, lSleepInt,
        IsLimitedAd, strIsBannerAd, lAdID, ContP, AdP, lAdID, ContP, AdP);
    *pCtxt << strHolderHTML;
    EndContent(pCtxt);
}

void CAdExtExtension::D(CHttpServerContext * pCtxt, LPCTSTR target, int AdNum,
    LPCTSTR etc)
{
    long lAdID=0;
    long lAdIDFirst=0;
    CString strServer;
    int ShowAd=1;
    char path[256];
    char port[10];

    IAdapterManager2Ptr AdMgr(__uuidof(AdManager2));
    int IsLimitedAd=AdMgr->IsLimitedAd;

```

```

//      if(!IsLimitedAd)
//          GetDinfo(pCtxt,AdNum,target,lAdID,lAdIDFirst,ShowAd,strServer);
/*      else
    {
        unsigned long len=256;
        pCtxt->GetServerVariable("SERVER_NAME", path, &len);
        pCtxt->GetServerVariable("SERVER_PORT", port, &len);
        strServer=path;
        CString strPort(port);

        if(strPort!="80")
        {
            strServer+=CString(":");
            strServer+=strPort;
        }
    }
*/

IFileOb2Ptr FileOb(__uuidof(FileOb2));

//          Set FileOb=Server.CreateObject("DualFrameComServer.FileOb")
//      CString strTarget=CString(buffer)+//+"&xxxxx=xxxxx&";
//      char path[256];
//      unsigned long len=256;
//      pCtxt->GetServerVariable("PATH_TRANSLATED ", path, &len);
//      CString strDHTMLfmt=LoadHTML(IDR_HTML_D);
//      CString strDHTML;
//      strDHTML.Format(strDHTMLfmt,lAdID,ShowAd==1,lAdID,lAdIDFirst,ShowAd==1);
//      *pCtxt << strDHTML;

//      long flag=0;
//      if (CString(etc)=="frame")
//          flag=1;
//      flag=IsLimitedAd;
//      CString strTarget(target);
//      strTarget+="&xxxxx=xxxxx&";
//      _bstr_t bstrServer(strServer);

//      _bstr_t strOut=FileOb-
//      >BuildContentPage(strTarget.AllocSysString(),AdNum,bstrServer,flag);
//      _bstr_t strOut=FileOb-
//      >BuildContentPage(CString(path).AllocSysString(),AdNum,bstrServer);
//      //      Response.write outstring
//      *pCtxt << strOut;

}
HRESULT CAdExtExtension::GetDinfo(CHttpServerContext * pCtxt,long AdNum,CString
target,long& lAdID,long& lAdIDFirst,int& ShowAd,CString& strServer)
{
//      TCHAR buffer[1025];

```

```

//      strcpy(buffer,target);
//      DWORD dwSize = 1024;
//      pCtxt->ServerSupportFunction(HSE_REQ_MAP_URL_TO_PATH, (void*)buffer,
&dwSize, NULL);

CString outstring;
CString      AdID("0");
CString filestr(target);
//      IHtmlBuilderObPtr HTMLob(__uuidof(HtmlBuilderOb));
//      HTMLob->PutURL(filestr.AllocSysString());
char path[256];
char port[10];

unsigned long len=256;
pCtxt->GetServerVariable("SERVER_NAME", path, &len);
pCtxt->GetServerVariable("SERVER_PORT", port, &len);
strServer=path;
CString strPort(port);

if(strPort!="80")
{
    strServer+=CString(":");
    strServer+=strPort;
}
_bstr_t bstrServer(strServer);

//      _bstr_t bstrServer=HTMLob->GetServer();
ShowAd=1;
#ifdef _NODB
    _RecordsetPtr  pRstLink("ADODB.Recordset");
    _ConnectionPtr pConn("ADODB.Connection");
    _CommandPtr pCmd(__uuidof(Command));
    pConn->Open("Provider=SQLOLEDB.1;Persist Security Info=False;User
ID=sa;Initial Catalog=DualFrame;Locale Identifier=1033;Connect Timeout=15;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;User Id=sa;", "",
"",
                adConnectUnspecified);
    pCmd->ActiveConnection=pConn;
    CString strSQLf="Select ID,HomeAddress from Ads where URL='%s' and
IsBanner=1";
    CString strSQL;
    strSQL.Format(strSQLf,filestr);
    pCmd->CommandText=strSQL.AllocSysString();
    pCmd->CommandType=adCmdText;
    pCmd->Prepared=true;
    _variant_t vr,vr1;
    vr=(long)0;
    vr1=(long)0;
    pRstLink=pCmd->Execute(&vr,&vr1,adOptionUnspecified);
    if(!pRstLink->EndOfFile)
        lAdID=(long) ( pRstLink->Fields->Item["ID"]->Value);
    IAdapterManagerPtr AdOb(__uuidof(AdManager2));
    AdOb->BuildAds(bstrServer);
    if (lAdID==0)
    {

```

```

        ShowAd=AdOb->GetAdAtPos(AdNum);
        lAdID=AdOb->AdID;
    }
    AdOb->GetLastAdPosition();
    lAdIDFirst=AdOb->AdID;
#endif
    return S_OK;
}
void CAdExtExtension::Ad(CHttpServerContext * pCtxt, long AdID)
{
    char path[256];
    char port[256];
    unsigned long len=256;
    pCtxt->GetServerVariable("SERVER_NAME", path, &len);
    pCtxt->GetServerVariable("SERVER_PORT", port, &len);
    CString strServer(path);
    CString strPort(port);

    if(strPort!="80")
    {
        strServer+=CString(":");
        strServer+=strPort;
    }

    CString strServer1("http://");
    strServer1+=strServer;
    IAdapterManagerPtr AdMgr(__uuidof(AdManager2));
    int IsLimitedAd=AdMgr->IsLimitedAd;

    IFileOb2Ptr FileOb(__uuidof(FileOb2));
    /*
    if(IsLimitedAd)
    {
        IAdapterManagerPtr AdMgr(__uuidof(AdManager2));
        _bstr_t bstrURLt=AdMgr->LimitedAdName;
        BSTR bstrURL1=bstrURLt.copy();
        CString strT1(bstrURL1);
        strT1+="&xxxxx=xxxxx&";
        _bstr_t strOut=FileOb->BuildAdPage(strT1.AllocSysString());
        *pCtxt << strOut;
        return;
    }*/
    _ConnectionPtr pConn("ADODB.Connection");
    pConn->Open("Provider=SQLOLEDB.1;Persist Security Info=False;User
ID=sa;Initial Catalog=DualFrame;Locale Identifier=1033;Connect Timeout=15;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;User Id=sa;", "",
"",
        adConnectUnspecified);
    _RecordsetPtr pRstLink("ADODB.Recordset");
    _CommandPtr pCmd(__uuidof(Command));
    pCmd->ActiveConnection=pConn;
    CString strSQLf="Select * from Ads where ID=%d";
    CString strSQL;

```

```

strSQL.Format(strSQLf,AdID);
pCmd->CommandText=strSQL.AllocSysString();
pCmd->CommandType=adCmdText;
pCmd->Prepared=true;
_variant_t vr,vr1;
vr=(long)0;
vr1=(long)0;
pRstLink=pCmd->Execute(&vr,&vr1,adOptionUnspecified);
CString strURL;
BOOL IsBanner=false;
CString strHome;
if(!pRstLink->EndOfFile)
{
    strURL=(char*) ((_bstr_t) pRstLink->Fields->Item["URL"]->Value);
    IsBanner=(long) ( pRstLink->Fields->Item["IsBanner"]->Value);
    strHome=(char*) ((_bstr_t) pRstLink->Fields->Item["HomeAddress"]-
>Value);
}
CString strDHTMLfmt=LoadHTML(IDR_HTML_AD);
CString strDHTML;

strDHTML.Format(strDHTMLfmt,strServer1.GetBuffer(0),AdID,strURL.GetBuffer
(0),strServer1.GetBuffer(0),AdID,strHome.GetBuffer(0));
*pCtxt << strDHTML;

if (!strURL.IsEmpty() && !IsBanner)
{
    CString strT(strURL+"&XXXXX=XXXXX&");
    _bstr_t strOut=FileOb->BuildAdPage(strT.AllocSysString());
    *pCtxt << strOut;
}
else if (IsBanner)
{
    strSQLf="Select
Alignment,Border,HorzSpacing,VertSpacing,Width,Height,MeasureW,MeasureH from
BannerAd where AdsId=%d";
    strSQL.Format(strSQLf,AdID);
    pCmd->CommandText=strSQL.AllocSysString();
    _RecordsetPtr pBannerAd("ADODB.Recordset");
    vr=(long)0;
    vr1=(long)0;
    pBannerAd=pCmd->Execute(&vr,&vr1,adOptionUnspecified);
    CString strAlign=(char*) ((_bstr_t) pBannerAd->Fields-
>Item["Alignment"]->Value);
    long border=(long) ( pBannerAd->Fields->Item["Border"]->Value);
    long HorzSpacing=(long) ( pBannerAd->Fields->Item["HorzSpacing"]-
>Value);
    long VertSpacing=(long) ( pBannerAd->Fields->Item["VertSpacing"]-
>Value);
    long width=(long) ( pBannerAd->Fields->Item["width"]->Value);
    long height=(long) ( pBannerAd->Fields->Item["height"]->Value);
    long MeasureW=(long) ( pBannerAd->Fields->Item["MeasureW"]->Value);
    long MeasureH=(long) ( pBannerAd->Fields->Item["MeasureH"]->Value);

    if (!IsServerExternal(pCtxt,strURL))

```

```

        strURL+="&xxxxx=xxxxx&";
        IRefCollection2Ptr Refs(__uuidof(RefCollection2));
        BSTR bs=Refs-
>BuildBannerAdFromHref(strHome.AllocSysString(),strURL.AllocSysString(),
        strAlign.AllocSysString(),border,HorzSpacing,VertSpacing,width,height,Mea
sureW,MeasureH);
        CString str1=bs;

        CString strFileSrc;
        char szBuffer[1024],szTempFileSrc[1024];
        DWORD result=::GetTempPath(1024,szBuffer);
        result=::GetTempFileName(szBuffer,"DFAD",0,szTempFileSrc);
        CStdioFile
fileSource(szTempFileSrc,CFile::modeCreate|CFile::modeReadWrite);
        int len=str1.GetLength();
        fileSource.Write(str1,len);
        char b[1];
        b[0]=0;
        fileSource.Write(&b,1);
        fileSource.Close();

        _bstr_t strOut=FileOb-
>BuildAdPage(CString(szTempFileSrc).AllocSysString());

        *pCtxt << strOut;
    }
//    ColeDateTime t=ColeDateTime::GetCurrentTime();

    try
    {
        _RecordsetPtr pSet("ADODB.Recordset");
        CComVariant vrSrc,vrConn,vrTime;
        ColeDateTime t;
        t=ColeDateTime::GetCurrentTime();
        vrSrc="Hits";
        vrConn.ChangeType(VT_DISPATCH);

        vrConn.pdispVal=pConn;
        vrTime.ChangeType(VT_DATE);
        vrTime.date=t;
        pSet->Open(vrSrc,_variant_t((IDispatch *)pConn,true),
adOpenKeyset,adLockOptimistic,adCmdTableDirect);
        pSet->AddNew();
        pSet->Fields->Item["AdID"]->Value=AdID;
        pSet->Fields->Item["HitType"]->Value=(long)3;
        pSet->Fields->Item["Time"]->Value=vrTime;
        pSet->Update();
        pSet->Close();
    }
    catch(...)
    {
    }

```

```

/*  CString strPath(path);
    strPath+=strURL;
    CString strHomePath(strPath);
    strHomePath+=strHome; */

/*

HitTime=Now
If infil <> "" and IsBanner = false then
    Response.write "4"
    Set FileOb=Server.CreateObject("DualFrameComServer.FileOb")
    Response.write "5"
    Response.write infil
    TempFile=FileOb.BuildAdPage(infil)
    Response.write infil

    set fso = Server.CreateObject("Scripting.FileSystemObject")
    Response.write "6"
    set fil = fso.OpenTextFile(TempFile,1)
    outstring = fil.ReadAll
    fil.Close
    fso.DeleteFile TempFile,true
    ' PRE tags preserve the format of the file
    Response.write outstring
End If
If infil <> "" and IsBanner = true then
    oCmd.CommandText="Select
Alignment,Border,HorzSpacing,VertSpacing,Width,Height,MeasureW,MeasureH from
BannerAd where AdsId="&AdID
    Set RecordsetBannerAd=oCmd.Execute
    Set
Refs=Server.CreateObject("DualFrameComServer.RefCollection")

    infil1=Refs.BuildBannerAdFromHref(recordset.Fields("HomeAddress"),infil,"
",RecordsetBannerAd.Fields("Alignment"),RecordsetBannerAd.Fields("Border"),Reco
rdsetBannerAd.Fields("HorzSpacing"),RecordsetBannerAd.Fields("VertSpacing"),Rec
ordsetBannerAd.Fields("width"),RecordsetBannerAd.Fields("height"),RecordsetBann
erAd.Fields("MeasureW"),RecordsetBannerAd.Fields("MeasureH"))
    Set FileOb=Server.CreateObject("DualFrameComServer.FileOb")
    TempFile=FileOb.BuildAdPage(infil1)
    set fso = Server.CreateObject("Scripting.FileSystemObject")
    set fil = fso.OpenTextFile(TempFile,1)
    outstring = fil.ReadAll
    fil.Close
    fso.DeleteFile TempFile,true
    fso.DeleteFile infil1,true
    ' PRE tags preserve the format of the file
    Response.write outstring
End If
Response.write "<script ID="&Chr(34) & "clientEventHandlersJS" & Chr(34) & "
LANGUAGE="& Chr(34) & "javascript"&Chr(34) & ">"&Chr(10)
'<script ID="clientEventHandlersJS" LANGUAGE="javascript">

```



```

Response.write "<!--"&Chr(10)
Response.write "function Hold_onclick() {"&Chr(10)
'Response.write "parent.open_window("&Chr(34) & recordset.Fields("URL")&
Chr(34) & ");"&Chr(10)
Response.write "parent.open_window
("&Chr(34)&"adhit.asp?HitType=2&ID="&AdID&"&HitTime="&HitTime&"&Page="&
recordset.Fields("URL") & Chr(34)&");"&Chr(10)

```

```

Response.write "}"&Chr(10)
Response.write "function HomePage_onclick() {"&Chr(10)
Response.write "parent.open_window
("&Chr(34)&"adhit.asp?HitType=1&ID="&AdID&"&HitTime="&HitTime&"&Page="&
recordset.Fields("HomeAddress") & Chr(34)&");"&Chr(10)
Response.write "}"&Chr(10)
Response.write "//-->"&Chr(10)
Response.write "</script>"

```

```

Set rst=Server.CreateObject("ADODB.Recordset")
rst.Open "Hits", obConn,1,3,512
'adOpenKeyset,adLockOptimistic,adCmdTableDirect
rst.AddNew
rst.Fields("AdID")=AdID
rst.Fields("HitType")=3 'SentToClient
rst.Fields("Time")=Now

```

```

rst.Update
rst.Close

```

```

'rst.Open "Select * from Ads where ID="&AdID, obConn,1,3
'adOpenKeyset,adLockOptimistic
'hitsStr=rst.Fields("SentToClient").Value
'hits=CInt(hitsStr)+1
'rst.Fields("SentToClient").Value = hits
'rst.Update
'rst.Close
Set oCmd.ActiveConnection=Nothing
'AdOb.SleepAd(2)
%>

```

```

*/

```

```

}

```

```

void CAdExtExtension::AdHit(CHttpServerContext * pCtxt, int HitType, long
lAdID,

```

```

LPCTSTR Page)

```

```

{

```

```

    try
    {

```

```

        _ConnectionPtr pConn("ADODB.Connection");
        pConn->Open("Provider=SQLOLEDB.1;Persist Security Info=False;User
ID=sa;Initial Catalog=DualFrame;Locale Identifier=1033;Connect Timeout=15;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;User Id=sa;", "",
"",

```

```

        adConnectUnspecified);

```

```

        _RecordsetPtr pSet("ADODB.Recordset");
        CComVariant vrSrc, vrConn, vrTime;
        COleDateTime t;
        t=COleDateTime::GetCurrentTime();
        vrSrc="Hits";
        vrConn.ChangeType(VT_DISPATCH);

        vrConn.pdispVal=pConn;
        vrTime.ChangeType(VT_DATE);
        vrTime.date=t;
        pSet->Open(vrSrc, _variant_t((IDispatch *)pConn, true),
adOpenKeyset, adLockOptimistic,      adCmdTableDirect);
        pSet->AddNew();
        pSet->Fields->Item["AdID"]->Value=lAdID;
        pSet->Fields->Item["HitType"]->Value=(long)HitType;
        pSet->Fields->Item["Time"]->Value=vrTime;
        pSet->Update();
        pSet->Close();
    }
    catch (_com_error &e)
    {
        TRACE((const char*)(e.Description()));
    }
    CString strPage(Page);
    if(!IsServerExternal(pCtxt, strPage))

        strPage+="&XXXXX=XXXXX&";
    if (HitType==2)
    {
        RedirectURL(pCtxt, strPage);
        return;
    }
    if (HitType==1)
    {
        RedirectURL(pCtxt, Page);
        return;
    }
}

/*

Set rst=Server.CreateObject("ADODB.Recordset")

rst.Open "Hits", obConn,1,3 ,512
'adOpenKeyset,adLockOptimistic,adCmdTableDirect
rst.AddNew
rst.Fields("AdID")=AdID
rst.Fields("HitType")=HitType
rst.Fields("Time")=HitTime

rst.Update
rst.Close
if HitType=2 then
    Set oCmd=Server.CreateObject("ADODB.Command")
    oCmd.ActiveConnection=Application("DualFrameDb_ConnectionString")
    oCmd.CommandText="Select * from Ads where ID=" &AdID
    oCmd.CommandType=1

```

```

oCmd.Prepared=true
set recordset=oCmd.Execute
infil=recordset.Fields("URL")

IsBanner=recordset.Fields("IsBanner")
If infil <> "" and IsBanner = false then
    set fso = Server.CreateObject("Scripting.FileSystemObject")
    set fil = fso.OpenTextFile(infil,1)
    outstring = fil.ReadAll
    fil.Close
    Response.write outstring
end if
If infil <> "" and IsBanner = true then
    oCmd.CommandText="Select
Alignment,Border,HorzSpacing,VertSpacing,Width,Height,MeasureW,MeasureH from
BannerAd where AdsId=&AdID
    Set RecordsetBannerAd=oCmd.Execute
    Set Refs=Server.CreateObject("DualFrameComServer.RefCollection")

    infil1=Refs.BuildBannerAdFromHref(recordset.Fields("HomeAddress"),infil,"
",RecordsetBannerAd.Fields("Alignment"),RecordsetBannerAd.Fields("Border"),Reco
rdsetBannerAd.Fields("HorzSpacing"),RecordsetBannerAd.Fields("VertSpacing"),Rec
ordsetBannerAd.Fields("width"),RecordsetBannerAd.Fields("height"),RecordsetBann
erAd.Fields("MeasureW"),RecordsetBannerAd.Fields("MeasureH"))

    Set Refs=Server.CreateObject("DualFrameComServer.RefCollection")
    infil1=Refs.BuildBannerAdFromHref(recordset.Fields("HomeAddress"),infil,"
")
    set fso = Server.CreateObject("Scripting.FileSystemObject")
    set fil = fso.OpenTextFile(infil1,1)
    outstring = fil.ReadAll
    fil.Close
    fso.DeleteFile infil1,true
    Response.write outstring
End If
else
    Response.Redirect Page
end if
*>
*/
}
bool CAdExtExtension::IsServerExternal(CHttpContext *pCtxt, CString
strURL)
{
    char path[256];
    bool IsExternal;

    unsigned long len=256;
    ASSERT(pCtxt);
    pCtxt->GetServerVariable("SERVER_NAME", path, &len);
    CString strActualServer(path);
    CHtmlEx obHT;
    CString strS=obHT.GetServer(strURL);
    if (strS==strActualServer)
        IsExternal=false;

```

```

        else
            IsExternal=true;
        return IsExternal;
    }
    CString CAdExtExtension::Encode(CHttpServerContext *pCtxt)
    {
        CString strBuf=pCtxt->ReadClient(
    }
    // Do not edit the following lines, which are needed by ClassWizard.
    #if 0
    BEGIN_MESSAGE_MAP(CAdExtExtension, CHttpServer)
        //{AFX_MSG_MAP(CAdExtExtension)
        //}AFX_MSG_MAP
    END_MESSAGE_MAP()
    #endif        // 0

////////////////////////////////////
// If your extension will not use MFC, you'll need this code to make
// sure the extension objects can find the resource handle for the
// module. If you convert your extension to not be dependent on MFC,
// remove the comments around the following AfxGetResourceHandle()
// and DllMain() functions, as well as the g_hInstance global.

/****

static HINSTANCE g_hInstance;

HINSTANCE AFXISAPI AfxGetResourceHandle()
{
    return g_hInstance;
}

BOOL WINAPI DllMain(HINSTANCE hInst, ULONG ulReason,
                    LPVOID lpReserved)
{
    if (ulReason == DLL_PROCESS_ATTACH)
    {
        g_hInstance = hInst;
    }

    return TRUE;
}

****/

```

APPENDIX 5c Adext.def

; ADEXT.def : declares the module parameters for the DLL.

LIBRARY "ADEXT"

EXPORTS

HttpExtensionProc  
GetExtensionVersion  
TerminateExtension

---

## APPENDIX 5d      Adext.dsp

```
# Microsoft Developer Studio Project File - Name="AdExt" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=AdExt - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "AdExt.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "AdExt.mak" CFG="AdExt - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "AdExt - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "AdExt - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "AdExt - Win32 Release"

# PROP BASE Use_MFC 2
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 2
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "Release"
# PROP Intermediate_Dir "Release"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
```

```

# ADD BASE CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_WINDLL" /D "_AFXDLL" /Yu"stdafx.h" /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_WINDLL" /D "_AFXDLL" /D "_MBCS" /D "_USRDLL" /Yu"stdafx.h" /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG" /d "_AFXDLL"
# ADD RSC /l 0x409 /d "NDEBUG" /d "_AFXDLL"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 /nologo /subsystem:windows /dll /machine:I386 /libpath:"..\..\lib"
# Begin Special Build Tool
OutDir=.\Release
SOURCE="$(InputPath)"
PostBuild_Cmds=copy $(OutDir)\AdExt.dll d:\inetpub\wwwroot\dpbin
# End Special Build Tool

!ELSEIF "$(CFG)" == "AdExt - Win32 Debug"

# PROP BASE Use_MFC 2
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 2
# PROP Use_Debug_Libraries 1
# PROP Output_Dir "Debug"
# PROP Intermediate_Dir "Debug"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /D "_WINDLL" /D "_AFXDLL" /Yu"stdafx.h" /FD /GZ /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/D "_WINDLL" /D "_AFXDLL" /D "_MBCS" /D "_USRDLL" /Yu"stdafx.h" /FD /GZ /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG" /d "_AFXDLL"
# ADD RSC /l 0x409 /d "_DEBUG" /d "_AFXDLL"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD LINK32 /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
/libpath:"..\..\lib"
# Begin Special Build Tool
OutDir=.\Debug
SOURCE="$(InputPath)"
PostBuild_Cmds=copy $(OutDir)\AdExt.dll d:\inetpub\wwwroot\dpbin
# End Special Build Tool

!ENDIF

```

```
# Begin Target

# Name "AdExt - Win32 Release"
# Name "AdExt - Win32 Debug"
# Begin Group "Source Files"

# PROP Default_Filter "cpp;c;cxx;rc;def;r;odl;idl;hpj;bat"
# Begin Source File

SOURCE=.\AdExt.cpp
# End Source File
# Begin Source File

SOURCE=.\AdExt.def
# End Source File
# Begin Source File

SOURCE=.\AdExt.rc
# End Source File
# Begin Source File

SOURCE=.\StdAfx.cpp
# ADD CPP /Yc"stdafx.h"
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "h;hpp;hxx;hm;inl"
# Begin Source File

SOURCE=.\AdExt.h
# End Source File
# Begin Source File

SOURCE=.\Resource.h
# End Source File
# Begin Source File

SOURCE=.\StdAfx.h
# End Source File
# End Group
# Begin Group "Resource Files"

# PROP Default_Filter "ico;cur;bmp;dlg;rc2;rct;bin;rgs;gif;jpg;jpeg;jpe"
# Begin Source File

SOURCE=.\AdExt.rc2
# End Source File
# Begin Source File

SOURCE=.\IDR_HTML_D_LAST.bin
# End Source File
# Begin Source File

SOURCE=.\IDR_HTML HOLDER_LAST.bin
# End Source File
# Begin Source File
```



```
SOURCE=.\IDR_HTML_HOLDER_ORIG.bin  
# End Source File  
# Begin Source File
```

```
SOURCE=.\IDR_HTMLorig.bin  
# End Source File  
# End Group  
# Begin Source File
```

```
SOURCE=.\html_ad.htm  
# End Source File  
# End Target  
# End Project
```

---

## APPENDIX 5e      Adext.h

```

#if !defined(AFX_ADEXT_H_F4430A9C_D282_11D3_8326_00400517BDC1__INCLUDED_)
#define AFX_ADEXT_H_F4430A9C_D282_11D3_8326_00400517BDC1__INCLUDED_

// ADEXT.H - Header file for your Internet Server
//      AdExt Extension

#include "resource.h"

class CAdExtExtension : public CHttpServer
{
public:
    CAdExtExtension();
    ~CAdExtExtension();
    void RedirectURL(CHttpServerContext *pCtxt, LPCTSTR url);
// Overrides
    // ClassWizard generated virtual function overrides
    // NOTE - the ClassWizard will add and remove member functions
here.
        //      DO NOT EDIT what you see in these blocks of generated code !
    //{AFX_VIRTUAL(CAdExtExtension)
    public:
    virtual BOOL GetExtensionVersion(HSE_VERSION_INFO* pVer);
    //}}AFX_VIRTUAL
    virtual BOOL TerminateExtension(DWORD dwFlags);

    // TODO: Add handlers for your commands here.
    // For example:

    void Default(CHttpServerContext* pCtxt);
    void Holder(CHttpServerContext* pCtxt, LPCTSTR target, LPCTSTR fvfvf);
    void D(CHttpServerContext * pCtxt, LPCTSTR target, int AdNum, LPCTSTR
etc);
    void Ad(CHttpServerContext * pCtxt, long AdID);
    void AdHit(CHttpServerContext * pCtxt, int HitType, long lAdID, LPCTSTR
Page);
    bool IsServerExternal(CHttpServerContext *pCtxt, CString strURL);
    HRESULT GetDinfo(CHttpServerContext * pCtxt, long AdNum, CString
target, long& lAdID, long& lAdIDFirst, int& ShowAd, CString& strServer);
    //      int m_IsLimitedAd;

    UINT m_nAdPercent;
    CString LoadHTML(UINT id);
    DECLARE_PARSE_MAP()

```

```
    //{AFX_MSG(CAdExtExtension)
    //}}AFX_MSG
};
```

```
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before
the previous line.
```

```
#endif // !defined(AFX_ADEXT_H__F4430A9C_D282_11D3_8326_00400517BDC1__INCLUDED)
```

## APPENDIX 5f      Adext.rc

```

//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include \"afxres.h\"\\r\\n"
    "\\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "#define _AFX_NO_SPLITTER_RESOURCES\\r\\n"

```

```

#define _AFX_NO_OLE_RESOURCES\r\n"
#define _AFX_NO_TRACKER_RESOURCES\r\n"
#define _AFX_NO_PROPERTY_RESOURCES\r\n"
\r\n"
\r\n"
#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)\r\n"
#ifdef _WIN32\r\n"
LANGUAGE 9, 1\r\n"
#pragma code_page(1252)\r\n"
#endif // _WIN32\r\n"
#include "AdExt.rc2" // non-Microsoft Visual C++ edited resources\r\n"
#include "afxres.rc" // Standard components\r\n"
#include "afxisapi.rc" // Internet Support resources\r\n"
#endif\r\n"
"\0"

END

#endif // APSTUDIO_INVOKED

#ifndef _MAC
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "\0"
            VALUE "CompanyName", "KazaGur\0"
            VALUE "FileDescription", "AdExt Internet Server Extension Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "ADEXT\0"
            VALUE "LegalCopyright", "Copyright (C) 2000 KazaGur\0"
            VALUE "LegalTrademarks", "\0"
            VALUE "OriginalFilename", "ADEXT.DLL\0"
            VALUE "PrivateBuild", "\0"
            VALUE "ProductName", "AdExt Internet Server Extension\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "SpecialBuild", "\0"
        END
    END
END
END

```

```
BLOCK "VarFileInfo"
BEGIN
    VALUE "Translation", 0x409, 1200
END
END

#endif    // !_MAC
```

```
////////////////////////////////////
//
// HTML
//
```

```
IDR_HTML_HOLDER HTML DISCARDABLE
BEGIN
```

```
0x0a0d, 0x733c, 0x7263, 0x7069, 0x2074, 0x414c, 0x474e, 0x4155, 0x4547,
0x273d, 0x736a, 0x7263, 0x7069, 0x2774, 0x0d3e, 0x3c0a, 0x2d21, 0x0d2d,
0x660a, 0x3d6c, 0x2522, 0x2273, 0x0d20, 0x2f0a, 0x2d2f, 0x3e2d, 0x0a0d,
0x6176, 0x2072, 0x7349, 0x6853, 0x776f, 0x6e4f, 0x6f4c, 0x6461, 0x253d,
0x3b64, 0x0a0d, 0x6176, 0x2072, 0x6177, 0x7469, 0x6e49, 0x6574, 0x7672,
0x6c61, 0x253d, 0x2a64, 0x3031, 0x3030, 0x0d3b, 0x760a, 0x7261, 0x4520,
0x7478, 0x7265, 0x616e, 0x536c, 0x7469, 0x4165, 0x4c64, 0x616f, 0x3d64,
0x6425, 0x0d3b, 0x760a, 0x7261, 0x4c20, 0x636f, 0x416b, 0x3d64, 0x6166,
0x736c, 0x3b65, 0x0a0d, 0x6176, 0x2072, 0x6f4c, 0x6461, 0x6441, 0x6c41,
0x6f6c, 0x6577, 0x3d64, 0x6166, 0x736c, 0x3b65, 0x0a0d, 0x6176, 0x2072,
0x7349, 0x6142, 0x6e6e, 0x7265, 0x6441, 0x253d, 0x3b73, 0x0a0d, 0x2f3c,
0x6373, 0x6972, 0x7470, 0x0d3e, 0x3c0a, 0x6373, 0x6972, 0x7470, 0x4c20,
0x4e41, 0x5547, 0x4741, 0x3d45, 0x6a22, 0x6373, 0x6972, 0x7470, 0x3e22,
0x0a0d, 0x213c, 0x2d2d, 0x0a0d, 0x6176, 0x2072, 0x6c66, 0x6761, 0x333d,
0x0d3b, 0x760a, 0x7261, 0x4920, 0x4273, 0x6765, 0x6e69, 0x743d, 0x7572,
0x3b65, 0x0a0d, 0x6176, 0x2072, 0x6441, 0x4449, 0x223d, 0x6425, 0x3b22,
0x0a0d, 0x6176, 0x2072, 0x6441, 0x4449, 0x6c4f, 0x3d64, 0x3b30, 0x0a0d,
0x0a0d, 0x7566, 0x636e, 0x6974, 0x6e6f, 0x6820, 0x6361, 0x6974, 0x6e6f,
0x6828, 0x6572, 0x5066, 0x7261, 0x6d61, 0x492c, 0x4573, 0x7478, 0x7265,
0x616e, 0x296c, 0x7b20, 0x0a0d, 0x6909, 0x2066, 0x7428, 0x706f, 0x452e,
0x7478, 0x7265, 0x616e, 0x536c, 0x7469, 0x4165, 0x4c64, 0x616f, 0x2964,
0x0a0d, 0x0909, 0x6f4c, 0x6b63, 0x6441, 0x213d, 0x7349, 0x7845, 0x6574,
0x6e72, 0x6c61, 0x0d3b, 0x090a, 0x6f4c, 0x6461, 0x6441, 0x6c41, 0x6f6c,
0x6577, 0x3d64, 0x7274, 0x6575, 0x093b, 0x0909, 0x2020, 0x2020, 0x0a0d,
0x6909, 0x2866, 0x4921, 0x4573, 0x7478, 0x7265, 0x616e, 0x296c, 0x0a0d,
0x0909, 0x6977, 0x646e, 0x776f, 0x742e, 0x706f, 0x742e, 0x706f, 0x2e31,
0x6f6c, 0x6163, 0x6974, 0x6e6f, 0x682e, 0x6572, 0x2066, 0x203d, 0x7268,
0x6665, 0x6150, 0x6172, 0x3b6d, 0x0a0d, 0x6509, 0x736c, 0x0d65, 0x090a,
0x7709, 0x6e69, 0x6f64, 0x2e77, 0x6f6c, 0x6163, 0x6974, 0x6e6f, 0x682e,
0x6572, 0x2066, 0x203d, 0x7268, 0x6665, 0x6150, 0x6172, 0x3b6d, 0x0a0d,
0x0d7d, 0x0d0a, 0x660a, 0x6e75, 0x7463, 0x6f69, 0x206e, 0x6f4c, 0x6461,
0x6843, 0x6c69, 0x4364, 0x6e6f, 0x2874, 0x7349, 0x6441, 0x0d29, 0x7b0a,
0x0a0d, 0x6909, 0x2866, 0x2828, 0x6f74, 0x2e70, 0x7349, 0x6542, 0x6967,
0x206e, 0x2626, 0x7420, 0x706f, 0x452e, 0x7478, 0x7265, 0x616e, 0x536c,
0x7469, 0x4165, 0x4c64, 0x616f, 0x2964, 0x7c20, 0x207c, 0x2128, 0x6f74,
0x2e70, 0x7845, 0x6574, 0x6e72, 0x6c61, 0x6953, 0x6574, 0x6441, 0x6f4c,
0x6461, 0x2929, 0x2620, 0x2026, 0x4928, 0x4173, 0x2064, 0x7c7c, 0x7420,
0x706f, 0x492e, 0x5373, 0x6f68, 0x4f77, 0x4c6e, 0x616f, 0x2964, 0x0d29,
0x090a, 0x0d7b, 0x090a, 0x7409, 0x706f, 0x642e, 0x636f, 0x6d75, 0x6e65,
0x2e74, 0x6f62, 0x7964, 0x722e, 0x776f, 0x3d73, 0x2522, 0x2c64, 0x6425,
0x0d22, 0x090a, 0x7409, 0x706f, 0x492e, 0x4273, 0x6765, 0x6e69, 0x663d,
```

0x6c61, 0x6573, 0x0a0d, 0x0909, 0x6977, 0x646e, 0x776f, 0x742e, 0x706f,  
0x622e, 0x746f, 0x6f74, 0x316d, 0x6c2e, 0x636f, 0x7461, 0x6f69, 0x2e6e,  
0x7268, 0x6665, 0x223d, 0x642f, 0x6270, 0x6e69, 0x412f, 0x4564, 0x7478,  
0x642e, 0x6c6c, 0x413f, 0x2664, 0x2b22, 0x6f74, 0x2e70, 0x6441, 0x4449,  
0x0d3b, 0x090a, 0x0d7d, 0x7d0a, 0x0d20, 0x0d0a, 0x660a, 0x6e75, 0x7463,  
0x6f69, 0x206e, 0x6f4c, 0x6461, 0x6946, 0x7372, 0x4374, 0x6e6f, 0x6574,  
0x746e, 0x2928, 0x0a0d, 0x0d7b, 0x090a, 0x7246, 0x6d61, 0x4e65, 0x6d61,  
0x3d65, 0x6c66, 0x0d3b, 0x090a, 0x6f74, 0x2e70, 0x7349, 0x6542, 0x6967,  
0x3d6e, 0x7274, 0x6575, 0x0a0d, 0x7709, 0x6e69, 0x6f64, 0x2e77, 0x6f74,  
0x2e70, 0x6f74, 0x3170, 0x6c2e, 0x636f, 0x7461, 0x6f69, 0x2e6e, 0x7268,  
0x6665, 0x3d20, 0x2220, 0x642f, 0x6270, 0x6e69, 0x412f, 0x4564, 0x7478,  
0x642e, 0x6c6c, 0x443f, 0x2226, 0x462b, 0x6172, 0x656d, 0x614e, 0x656d,  
0x222b, 0x3026, 0x2226, 0x0d3b, 0x090a, 0x6f74, 0x2e70, 0x6441, 0x4449,  
0x223d, 0x6425, 0x3b22, 0x0a0d, 0x0d7d, 0x0d0a, 0x660a, 0x6e75, 0x7463,  
0x6f69, 0x206e, 0x6f4c, 0x6461, 0x6843, 0x6c69, 0x2864, 0x7349, 0x6441,  
0x0d29, 0x7b0a, 0x0a0d, 0x6909, 0x2066, 0x7428, 0x706f, 0x492e, 0x4273,  
0x6765, 0x6e69, 0x3d21, 0x7274, 0x6575, 0x0d29, 0x090a, 0x7409, 0x6d69,  
0x6f65, 0x7475, 0x6176, 0x3d72, 0x6177, 0x7469, 0x6e49, 0x6574, 0x7672,  
0x6c61, 0x0d3b, 0x090a, 0x6c65, 0x6573, 0x0a0d, 0x0909, 0x6974, 0x656d,  
0x756f, 0x7674, 0x7261, 0x303d, 0x0d3b, 0x090a, 0x6669, 0x2128, 0x6f74,  
0x2e70, 0x7349, 0x6853, 0x776f, 0x6e4f, 0x6f4c, 0x6461, 0x7c20, 0x207c,  
0x6f4c, 0x6b63, 0x6441, 0x0d29, 0x090a, 0x7409, 0x6d69, 0x6f65, 0x7475,  
0x6176, 0x3d72, 0x3b30, 0x0a0d, 0x6909, 0x2066, 0x2828, 0x6f74, 0x2e70,  
0x7349, 0x6542, 0x6967, 0x3d6e, 0x743d, 0x7572, 0x2065, 0x7c7c, 0x7420,  
0x706f, 0x492e, 0x5373, 0x6f68, 0x4f77, 0x4c6e, 0x616f, 0x2964, 0x0d29,  
0x090a, 0x7709, 0x6e69, 0x6f64, 0x2e77, 0x6573, 0x5474, 0x6d69, 0x6f65,  
0x7475, 0x2228, 0x6f4c, 0x6461, 0x6843, 0x6c69, 0x4364, 0x6e6f, 0x2874,  
0x2b22, 0x7349, 0x6441, 0x222b, 0x2229, 0x742c, 0x6d69, 0x6f65, 0x7475,  
0x6176, 0x2972, 0x0d3b, 0x7d0a, 0x0a0d, 0x0a0d, 0x0a0d, 0x0a0d, 0x0a0d,  
0x7566, 0x636e, 0x6974, 0x6e6f, 0x4c20, 0x616f, 0x4164, 0x2864, 0x7349,  
0x6441, 0x0d29, 0x7b0a, 0x0a0d, 0x6909, 0x2066, 0x2128, 0x6f4c, 0x6461,  
0x6441, 0x6c41, 0x6f6c, 0x6577, 0x2964, 0x0a0d, 0x0909, 0x6572, 0x7574,  
0x6e72, 0x0d3b, 0x090a, 0x6669, 0x4928, 0x4173, 0x2064, 0x7c7c, 0x7420,  
0x706f, 0x492e, 0x5373, 0x6f68, 0x4f77, 0x4c6e, 0x616f, 0x2964, 0x0a0d,  
0x7b09, 0x0a0d, 0x0909, 0x6176, 0x2072, 0x7266, 0x3d6d, 0x6f74, 0x2e70,  
0x6f64, 0x7563, 0x656d, 0x746e, 0x662e, 0x6172, 0x656d, 0x3b73, 0x0a0d,  
0x0909, 0x6176, 0x2072, 0x3170, 0x663d, 0x6d72, 0x3128, 0x3b29, 0x0a0d,  
0x0909, 0x6669, 0x2820, 0x4c21, 0x636f, 0x416b, 0x2964, 0x0a0d, 0x0909,  
0x0d7b, 0x090a, 0x0909, 0x6f74, 0x2e70, 0x6f64, 0x7563, 0x656d, 0x746e,  
0x612e, 0x6c6c, 0x742e, 0x6761, 0x2873, 0x6622, 0x6172, 0x656d, 0x6573,  
0x2274, 0x2e29, 0x7469, 0x6d65, 0x3028, 0x2e29, 0x6f72, 0x7377, 0x223d,  
0x2c30, 0x3031, 0x3b22, 0x0a0d, 0x0909, 0x6909, 0x2866, 0x6f74, 0x2e70,  
0x7845, 0x6574, 0x6e72, 0x6c61, 0x6953, 0x6574, 0x6441, 0x6f4c, 0x6461,  
0x0d29, 0x090a, 0x0909, 0x0d7b, 0x090a, 0x0909, 0x2f09, 0x6d2f, 0x7779,  
0x6e69, 0x3d20, 0x7720, 0x6e69, 0x6f64, 0x2e77, 0x706f, 0x6e65, 0x2228,  
0x6441, 0x7845, 0x2e74, 0x6c64, 0x3f6c, 0x6441, 0x2226, 0x742b, 0x706f,  
0x412e, 0x4964, 0x4f44, 0x646c, 0x222c, 0x6977, 0x226e, 0x272c, 0x6977,  
0x7464, 0x3d68, 0x2c31, 0x6568, 0x6769, 0x7468, 0x313d, 0x6c2c, 0x6665,  
0x3d74, 0x3034, 0x3030, 0x742c, 0x706f, 0x343d, 0x3030, 0x2c30, 0x6974,  
0x6c74, 0x6265, 0x7261, 0x303d, 0x742c, 0x6f6f, 0x626c, 0x7261, 0x303d,  
0x6c2c, 0x636f, 0x7461, 0x6f69, 0x3d6e, 0x2c30, 0x6964, 0x6572, 0x7463,  
0x726f, 0x6569, 0x3d73, 0x2c30, 0x7473, 0x7461, 0x7375, 0x303d, 0x6d2c,  
0x6e65, 0x6275, 0x7261, 0x303d, 0x732c, 0x7263, 0x6c6f, 0x626c, 0x7261,  
0x3d73, 0x2c31, 0x6572, 0x6973, 0x617a, 0x6c62, 0x3d65, 0x2731, 0x312c,  
0x3b29, 0x0a0d, 0x0909, 0x0909, 0x6977, 0x646e, 0x776f, 0x732e, 0x7465,  
0x6954, 0x656d, 0x756f, 0x2874, 0x7322, 0x6d6f, 0x2865, 0x2229, 0x312c,  
0x3b29, 0x0a0d, 0x0909, 0x7d09, 0x0a0d, 0x2f2f, 0x0909, 0x0909, 0x706f,

0x6e65, 0x775f, 0x6e69, 0x6f64, 0x2877, 0x4122, 0x4564, 0x7478, 0x642e,  
0x6c6c, 0x413f, 0x2664, 0x2b22, 0x6f74, 0x2e70, 0x6441, 0x4449, 0x6c4f,  
0x2964, 0x0920, 0x0a0d, 0x2f2f, 0x0909, 0x0909, 0x6977, 0x646e, 0x776f,  
0x732e, 0x7465, 0x6954, 0x656d, 0x756f, 0x2874, 0x6422, 0x6c65, 0x7961,  
0x7728, 0x6961, 0x4974, 0x746e, 0x7265, 0x6176, 0x296c, 0x2c22, 0x2930,  
0x0d3b, 0x090a, 0x2f09, 0x092f, 0x6c61, 0x7265, 0x2874, 0x6f27, 0x276f,  
0x0d29, 0x0d0a, 0x090a, 0x0909, 0x0a0d, 0x0909, 0x0a0d, 0x0909, 0x0d7d,  
0x090a, 0x0d7d, 0x090a, 0x6974, 0x656d, 0x756f, 0x7674, 0x7261, 0x773d,  
0x6961, 0x4974, 0x746e, 0x7265, 0x6176, 0x3b6c, 0x0a0d, 0x6909, 0x2866,  
0x6f4c, 0x6b63, 0x6441, 0x0d29, 0x090a, 0x7409, 0x6d69, 0x6f65, 0x7475,  
0x6176, 0x3d72, 0x3b30, 0x0a0d, 0x4c09, 0x616f, 0x4164, 0x4164, 0x6c6c,  
0x776f, 0x6465, 0x663d, 0x6c61, 0x6573, 0x0d3b, 0x090a, 0x6669, 0x4928,  
0x4173, 0x2064, 0x2626, 0x2820, 0x7421, 0x706f, 0x492e, 0x5373, 0x6f68,  
0x4f77, 0x4c6e, 0x616f, 0x2964, 0x0d29, 0x090a, 0x0d7b, 0x090a, 0x7709,  
0x6e69, 0x6f64, 0x2e77, 0x6573, 0x5474, 0x6d69, 0x6f65, 0x7475, 0x2228,  
0x6f4c, 0x6461, 0x6843, 0x6c69, 0x4364, 0x6e6f, 0x2874, 0x2b22, 0x7349,  
0x6441, 0x222b, 0x2229, 0x742c, 0x6d69, 0x6f65, 0x7475, 0x6176, 0x2972,  
0x0d3b, 0x090a, 0x0d7d, 0x090a, 0x0a0d, 0x0d7d, 0x0d0a, 0x0d0a, 0x660a,  
0x6e75, 0x7463, 0x6f69, 0x206e, 0x706f, 0x6e65, 0x775f, 0x6e69, 0x6f64,  
0x2877, 0x7275, 0x296c, 0x7b20, 0x0a0d, 0x796d, 0x6977, 0x206e, 0x203d,  
0x6977, 0x646e, 0x776f, 0x6f2e, 0x6570, 0x286e, 0x7275, 0x2c6c, 0x7722,  
0x6e69, 0x2c22, 0x7427, 0x7469, 0x656c, 0x6162, 0x3d72, 0x2c30, 0x6f74,  
0x6c6f, 0x6162, 0x3d72, 0x2c30, 0x6f6c, 0x6163, 0x6974, 0x6e6f, 0x303d,  
0x642c, 0x7269, 0x6365, 0x6f74, 0x6972, 0x7365, 0x303d, 0x732c, 0x6174,  
0x7574, 0x3d73, 0x2c30, 0x656d, 0x756e, 0x6162, 0x3d72, 0x2c30, 0x6373,  
0x6f72, 0x6c6c, 0x6162, 0x7372, 0x313d, 0x722c, 0x7365, 0x7a69, 0x6261,  
0x656c, 0x313d, 0x772c, 0x6469, 0x6874, 0x353d, 0x3634, 0x682c, 0x6965,  
0x6867, 0x3d74, 0x3035, 0x2730, 0x312c, 0x3b29, 0x0a0d, 0x0d7d, 0x2f0a,  
0x2d2f, 0x3e2d, 0x0a0d, 0x2f3c, 0x6373, 0x6972, 0x7470, 0x0d3e, 0x3c0a,  
0x6373, 0x6972, 0x7470, 0x4c20, 0x4e41, 0x5547, 0x4741, 0x3d45, 0x6a22,  
0x6373, 0x6972, 0x7470, 0x3e22, 0x0a0d, 0x213c, 0x2d2d, 0x0a0d, 0x7566,  
0x636e, 0x6974, 0x6e6f, 0x6420, 0x6c65, 0x7961, 0x6428, 0x6c65, 0x0d29,  
0x7b0a, 0x0a0d, 0x7609, 0x7261, 0x6420, 0x3d20, 0x6e20, 0x7765, 0x4420,  
0x7461, 0x2865, 0x3b29, 0x0a0d, 0x2020, 0x2020, 0x6176, 0x2072, 0x736d,  
0x3d20, 0x6420, 0x672e, 0x7465, 0x6954, 0x656d, 0x2928, 0x0d3b, 0x090a,  
0x0a0d, 0x6109, 0x303d, 0x0d3b, 0x090a, 0x6f66, 0x2072, 0x6928, 0x303d,  
0x693b, 0x313c, 0x3030, 0x3030, 0x3030, 0x3030, 0x693b, 0x2b2b, 0x0d29,  
0x090a, 0x0d7b, 0x090a, 0x7609, 0x7261, 0x6420, 0x2031, 0x203d, 0x656e,  
0x2077, 0x6144, 0x6574, 0x2928, 0x0d3b, 0x090a, 0x6909, 0x2066, 0x6428,  
0x2e31, 0x6567, 0x5474, 0x6d69, 0x2865, 0x3e29, 0x736d, 0x642b, 0x6c65,  
0x0d29, 0x090a, 0x0909, 0x7262, 0x6165, 0x3b6b, 0x0a0d, 0x0909, 0x6977,  
0x646e, 0x776f, 0x732e, 0x7465, 0x6954, 0x656d, 0x756f, 0x2874, 0x7322,  
0x6d6f, 0x2865, 0x2229, 0x312c, 0x3030, 0x2930, 0x0d3b, 0x090a, 0x097d,  
0x0a0d, 0x0a0d, 0x0d7d, 0x660a, 0x6e75, 0x7463, 0x6f69, 0x206e, 0x6f73,  
0x656d, 0x2928, 0x0a0d, 0x0d7b, 0x090a, 0x796d, 0x6977, 0x206e, 0x203d,  
0x6977, 0x646e, 0x776f, 0x6f2e, 0x6570, 0x286e, 0x2f22, 0x7064, 0x6962,  
0x2f6e, 0x6441, 0x7845, 0x2e74, 0x6c64, 0x3f6c, 0x6441, 0x2226, 0x742b,  
0x706f, 0x412e, 0x4964, 0x4f44, 0x646c, 0x222c, 0x6977, 0x222e, 0x272c,  
0x6977, 0x7464, 0x3d68, 0x2c31, 0x6568, 0x6769, 0x7468, 0x313d, 0x6c2c,  
0x6665, 0x3d74, 0x3034, 0x3030, 0x742c, 0x706f, 0x343d, 0x3030, 0x2c30,  
0x6974, 0x6c74, 0x6265, 0x7261, 0x303d, 0x742c, 0x6f6f, 0x626c, 0x7261,  
0x303d, 0x6c2c, 0x636f, 0x7461, 0x6f69, 0x3d6e, 0x2c30, 0x6964, 0x6572,  
0x7463, 0x726f, 0x6569, 0x3d73, 0x2c30, 0x7473, 0x7461, 0x7375, 0x303d,  
0x6d2c, 0x6e65, 0x6275, 0x7261, 0x303d, 0x732c, 0x7263, 0x6c6f, 0x626c,  
0x7261, 0x3d73, 0x2c31, 0x6572, 0x6973, 0x617a, 0x6c62, 0x3d65, 0x2731,  
0x312c, 0x3b29, 0x0a0d, 0x2f2f, 0x6d09, 0x7779, 0x6e69, 0x3d20, 0x7720,  
0x6e69, 0x6f64, 0x2e77, 0x706f, 0x6e65, 0x2228, 0x642f, 0x6270, 0x6e69,



0x412f, 0x4564, 0x7478, 0x642e, 0x6c6c, 0x413f, 0x2664, 0x2b22, 0x6f74,  
0x2e70, 0x6441, 0x4449, 0x6c4f, 0x2c64, 0x7722, 0x6e69, 0x2c22, 0x7727,  
0x6469, 0x6874, 0x313d, 0x682c, 0x6965, 0x6867, 0x3d74, 0x2c31, 0x656c,  
0x7466, 0x343d, 0x3030, 0x2c30, 0x6f74, 0x3d70, 0x3034, 0x3030, 0x742c,  
0x7469, 0x656c, 0x6162, 0x3d72, 0x2c31, 0x6f74, 0x6c6f, 0x6162, 0x3d72,  
0x2c31, 0x6f6c, 0x6163, 0x6974, 0x6e6f, 0x313d, 0x642c, 0x7269, 0x6365,  
0x6f74, 0x6972, 0x7365, 0x313d, 0x732c, 0x6174, 0x7574, 0x3d73, 0x2c31,  
0x656d, 0x756e, 0x6162, 0x3d72, 0x2c31, 0x6373, 0x6f72, 0x6c6c, 0x6162,  
0x7372, 0x313d, 0x722c, 0x7365, 0x7a69, 0x6261, 0x656c, 0x313d, 0x2c27,  
0x2931, 0x0d3b, 0x090a, 0x796d, 0x6977, 0x2e6e, 0x6c62, 0x7275, 0x2928,  
0x0d3b, 0x090a, 0x6977, 0x646e, 0x776f, 0x662e, 0x636f, 0x7375, 0x2928,  
0x0d3b, 0x090a, 0x796d, 0x6977, 0x2e6e, 0x6572, 0x6973, 0x657a, 0x6f54,  
0x3528, 0x3634, 0x202c, 0x3035, 0x2930, 0x0d3b, 0x090a, 0x796d, 0x6977,  
0x2e6e, 0x6f6d, 0x6576, 0x6f54, 0x3128, 0x3030, 0x202c, 0x3031, 0x2930,  
0x0a0d, 0x7274, 0x0d79, 0x7b0a, 0x0a0d, 0x6d09, 0x7779, 0x6e69, 0x662e,  
0x616c, 0x6873, 0x626f, 0x532e, 0x6f74, 0x2870, 0x3b29, 0x0a0d, 0x6d09,  
0x7779, 0x6e69, 0x662e, 0x616c, 0x6873, 0x626f, 0x462e, 0x6172, 0x656d,  
0x754e, 0x3d6d, 0x3031, 0x0d3b, 0x090a, 0x796d, 0x6977, 0x2e6e, 0x6c62,  
0x7275, 0x2928, 0x0d3b, 0x090a, 0x796d, 0x6977, 0x2e6e, 0x794d, 0x7546,  
0x636e, 0x6974, 0x6e6f, 0x7728, 0x6e69, 0x6f64, 0x2977, 0x0d3b, 0x0d0a,  
0x090a, 0x0d7d, 0x630a, 0x7461, 0x6863, 0x6528, 0x0d29, 0x7b0a, 0x0a0d,  
0x0d7d, 0x0d0a, 0x7d0a, 0x0a0d, 0x0a0d, 0x7566, 0x636e, 0x6974, 0x6e6f,  
0x6220, 0x6361, 0x6974, 0x6e6f, 0x6828, 0x6572, 0x5066, 0x7261, 0x6d61,  
0x492c, 0x4573, 0x7478, 0x7265, 0x616e, 0x296c, 0x7b20, 0x0a0d, 0x6909,  
0x2066, 0x7428, 0x706f, 0x452e, 0x7478, 0x7265, 0x616e, 0x536c, 0x7469,  
0x4165, 0x4c64, 0x616f, 0x2964, 0x0a0d, 0x0909, 0x6f4c, 0x6461, 0x6441,  
0x6c41, 0x6f6c, 0x6577, 0x3d64, 0x7274, 0x6575, 0x093b, 0x0909, 0x2020,  
0x2020, 0x0a0d, 0x6909, 0x2866, 0x4921, 0x4573, 0x7478, 0x7265, 0x616e,  
0x296c, 0x0a0d, 0x0909, 0x6977, 0x646e, 0x776f, 0x6c2e, 0x636f, 0x7461,  
0x6f69, 0x2e6e, 0x7268, 0x6665, 0x3d20, 0x6820, 0x6572, 0x5066, 0x7261,  
0x6d61, 0x0d3b, 0x090a, 0x6c65, 0x6573, 0x0a0d, 0x0909, 0x6977, 0x646e,  
0x776f, 0x742e, 0x706f, 0x742e, 0x706f, 0x2e31, 0x6f6c, 0x6163, 0x6974,  
0x6e6f, 0x682e, 0x6572, 0x2066, 0x203d, 0x7268, 0x6665, 0x6150, 0x6172,  
0x3b6d, 0x0a0d, 0x0d7d, 0x0d0a, 0x660a, 0x6e75, 0x7463, 0x6f69, 0x206e,  
0x6552, 0x7266, 0x7365, 0x4368, 0x6968, 0x646c, 0x6f43, 0x746e, 0x2928,  
0x0a0d, 0x0d7b, 0x090a, 0x6669, 0x2828, 0x7428, 0x706f, 0x492e, 0x4273,  
0x6765, 0x6e69, 0x2620, 0x2026, 0x6f74, 0x2e70, 0x7845, 0x6574, 0x6e72,  
0x6c61, 0x6953, 0x6574, 0x6441, 0x6f4c, 0x6461, 0x2029, 0x7c7c, 0x2820,  
0x7421, 0x706f, 0x452e, 0x7478, 0x7265, 0x616e, 0x536c, 0x7469, 0x4165,  
0x4c64, 0x616f, 0x2964, 0x2029, 0x2626, 0x2820, 0x7349, 0x6441, 0x7c20,  
0x207c, 0x6f74, 0x2e70, 0x7349, 0x6853, 0x776f, 0x6e4f, 0x6f4c, 0x6461,  
0x2929, 0x0a0d, 0x7b09, 0x0a0d, 0x0909, 0x6f74, 0x2e70, 0x6f64, 0x7563,  
0x656d, 0x746e, 0x622e, 0x646f, 0x2e79, 0x6f72, 0x7377, 0x223d, 0x2c30,  
0x3031, 0x0d22, 0x090a, 0x7409, 0x706f, 0x492e, 0x4273, 0x6765, 0x6e69,  
0x743d, 0x7572, 0x0d65, 0x090a, 0x7709, 0x6e69, 0x6f64, 0x2e77, 0x6f74,  
0x2e70, 0x6f62, 0x7474, 0x6d6f, 0x2e31, 0x6f6c, 0x6163, 0x6974, 0x6e6f,  
0x682e, 0x6572, 0x3d66, 0x2f22, 0x7064, 0x6962, 0x2f6e, 0x6441, 0x7845,  
0x2e74, 0x6c64, 0x3f6c, 0x2644, 0x2b22, 0x6f74, 0x2e70, 0x6441, 0x4449,  
0x0d3b, 0x090a, 0x0d7d, 0x7d0a, 0x0d20, 0x0d0a, 0x660a, 0x6e75, 0x7463,  
0x6f69, 0x206e, 0x6f4c, 0x6461, 0x6553, 0x6f63, 0x646e, 0x6f43, 0x746e,  
0x6e65, 0x2874, 0x0d29, 0x7b0a, 0x0a0d, 0x4609, 0x6172, 0x656d, 0x614e,  
0x656d, 0x663d, 0x3b6c, 0x0a0d, 0x7409, 0x706f, 0x492e, 0x4273, 0x6765,  
0x6e69, 0x743d, 0x7572, 0x0d65, 0x090a, 0x6977, 0x646e, 0x776f, 0x742e,  
0x706f, 0x742e, 0x706f, 0x2e31, 0x6f6c, 0x6163, 0x6974, 0x6e6f, 0x682e,  
0x6572, 0x2066, 0x203d, 0x2f22, 0x7064, 0x6962, 0x2f6e, 0x6441, 0x7845,  
0x2e74, 0x6c64, 0x3f6c, 0x2644, 0x2b22, 0x7246, 0x6d61, 0x4e65, 0x6d61,  
0x2b65, 0x2622, 0x2630, 0x3b22, 0x0a0d, 0x7409, 0x706f, 0x412e, 0x4964,

0x3d44, 0x3322, 0x3b22, 0x0a0d, 0x0d7d, 0x0d0a, 0x660a, 0x6e75, 0x7463,  
0x6f69, 0x206e, 0x6c50, 0x7961, 0x6441, 0x4928, 0x4173, 0x2964, 0x0a0d,  
0x0d7b, 0x090a, 0x6669, 0x2820, 0x6f74, 0x2e70, 0x7349, 0x6542, 0x6967,  
0x3d6e, 0x743d, 0x7572, 0x2965, 0x0a0d, 0x0909, 0x6974, 0x656d, 0x756f,  
0x7674, 0x7261, 0x773d, 0x6961, 0x4974, 0x746e, 0x7265, 0x6176, 0x3b6c,  
0x0a0d, 0x6509, 0x736c, 0x0d65, 0x090a, 0x7409, 0x6d69, 0x6f65, 0x7475,  
0x6176, 0x3d72, 0x3b30, 0x0a0d, 0x6909, 0x2866, 0x6f74, 0x2e70, 0x7349,  
0x6853, 0x776f, 0x6e4f, 0x6f4c, 0x6461, 0x2620, 0x2026, 0x6f4c, 0x6b63,  
0x6441, 0x0d29, 0x090a, 0x7409, 0x6d69, 0x6f65, 0x7475, 0x6176, 0x3d72,  
0x3b30, 0x0a0d, 0x6909, 0x2066, 0x2828, 0x6f74, 0x2e70, 0x7349, 0x6542,  
0x6967, 0x3d6e, 0x743d, 0x7572, 0x2065, 0x7c7c, 0x7420, 0x706f, 0x492e,  
0x5373, 0x6f68, 0x4f77, 0x4c6e, 0x616f, 0x2964, 0x0d29, 0x090a, 0x7709,  
0x6e69, 0x6f64, 0x2e77, 0x6573, 0x5474, 0x6d69, 0x6f65, 0x7475, 0x2228,  
0x6552, 0x7266, 0x7365, 0x4368, 0x6968, 0x646c, 0x6f43, 0x746e, 0x2228,  
0x492b, 0x4173, 0x2b64, 0x2922, 0x2c22, 0x6974, 0x656d, 0x756f, 0x7674,  
0x7261, 0x3b29, 0x0a0d, 0x0d7d, 0x0d0a, 0x0d0a, 0x0d0a, 0x660a,  
0x6e75, 0x7463, 0x6f69, 0x206e, 0x6853, 0x7475, 0x6441, 0x4928, 0x4173,  
0x2964, 0x0a0d, 0x0d7b, 0x090a, 0x6669, 0x2820, 0x6f4c, 0x6461, 0x6441,  
0x6c41, 0x6f6c, 0x6577, 0x2964, 0x0a0d, 0x0909, 0x6c50, 0x7961, 0x6441,  
0x4928, 0x4173, 0x2964, 0x0d3b, 0x0d0a, 0x090a, 0x6669, 0x4928, 0x4173,  
0x2064, 0x2626, 0x7420, 0x706f, 0x492e, 0x5373, 0x6f68, 0x4f77, 0x4c6e,  
0x616f, 0x2964, 0x0a0d, 0x7b09, 0x0a0d, 0x0909, 0x6176, 0x2072, 0x7266,  
0x3d6d, 0x6f74, 0x2e70, 0x6f64, 0x7563, 0x656d, 0x746e, 0x662e, 0x6172,  
0x656d, 0x3b73, 0x0a0d, 0x0909, 0x6176, 0x2072, 0x3170, 0x663d, 0x6d72,  
0x3128, 0x3b29, 0x0a0d, 0x0909, 0x6669, 0x2820, 0x4c21, 0x636f, 0x416b,  
0x2964, 0x0a0d, 0x0909, 0x7409, 0x706f, 0x642e, 0x636f, 0x6d75, 0x6e65,  
0x2e74, 0x6c61, 0x2e6c, 0x6174, 0x7367, 0x2228, 0x7266, 0x6d61, 0x7365,  
0x7465, 0x2922, 0x692e, 0x6574, 0x286d, 0x2930, 0x722e, 0x776f, 0x3d73,  
0x3022, 0x312c, 0x2230, 0x0d3b, 0x090a, 0x0d7d, 0x090a, 0x6974, 0x656d,  
0x756f, 0x7674, 0x7261, 0x773d, 0x6961, 0x4974, 0x746e, 0x7265, 0x6176,  
0x3b6c, 0x0a0d, 0x6909, 0x2866, 0x4c21, 0x636f, 0x416b, 0x2964, 0x0a0d,  
0x0909, 0x6f4c, 0x6461, 0x6441, 0x6c41, 0x6f6c, 0x6577, 0x3d64, 0x6166,  
0x736c, 0x3b65, 0x0a0d, 0x6909, 0x2866, 0x7349, 0x6441, 0x2620, 0x2026,  
0x2128, 0x6f74, 0x2e70, 0x7349, 0x6853, 0x776f, 0x6e4f, 0x6f4c, 0x6461,  
0x2929, 0x0a0d, 0x7b09, 0x0a0d, 0x0909, 0x6977, 0x646e, 0x776f, 0x732e,  
0x7465, 0x6954, 0x656d, 0x756f, 0x2874, 0x5022, 0x616c, 0x4179, 0x2864,  
0x2b22, 0x7349, 0x6441, 0x222b, 0x2229, 0x742c, 0x6d69, 0x6f65, 0x7475,  
0x6176, 0x2972, 0x0d3b, 0x090a, 0x0d7d, 0x090a, 0x0a0d, 0x0d7d, 0x0d0a,  
0x0d0a, 0x0d0a, 0x2f0a, 0x2d2f, 0x3e2d, 0x0a0d, 0x2f3c, 0x6373, 0x6972,  
0x7470, 0x0d3e, 0x0d0a, 0x3c0a, 0x7468, 0x6c6d, 0x0d3e, 0x3c0a, 0x6568,  
0x6461, 0x0d3e, 0x3c0a, 0x6974, 0x6c74, 0x3e65, 0x6f68, 0x646c, 0x7265,  
0x2f3c, 0x6974, 0x6c74, 0x3e65, 0x0a0d, 0x6d3c, 0x7465, 0x2061, 0x616e,  
0x656d, 0x223d, 0x4547, 0x454e, 0x4152, 0x4f54, 0x2252, 0x6320, 0x6e6f,  
0x6574, 0x746e, 0x223d, 0x694d, 0x7263, 0x736f, 0x666f, 0x2074, 0x7246,  
0x6e6f, 0x5074, 0x6761, 0x2065, 0x2e34, 0x2230, 0x0d3e, 0x3c0a, 0x656d,  
0x6174, 0x6e20, 0x6d61, 0x3d65, 0x5022, 0x6f72, 0x4967, 0x2264, 0x6320,  
0x6e6f, 0x6574, 0x746e, 0x223d, 0x7246, 0x6e6f, 0x5074, 0x6761, 0x2e65,  
0x6445, 0x7469, 0x726f, 0x442e, 0x636f, 0x6d75, 0x6e65, 0x2274, 0x0d3e,  
0x3c0a, 0x682f, 0x6165, 0x3e64, 0x0a0d, 0x663c, 0x6172, 0x656d, 0x6573,  
0x2074, 0x6f72, 0x7377, 0x223d, 0x6425, 0x252c, 0x2264, 0x6620, 0x6172,  
0x656d, 0x6f62, 0x6472, 0x7265, 0x223d, 0x2230, 0x6620, 0x6172, 0x656d,  
0x7073, 0x6361, 0x6e69, 0x3d67, 0x3022, 0x3e22, 0x0a0d, 0x2020, 0x663c,  
0x6172, 0x656d, 0x6e20, 0x6d61, 0x3d65, 0x7422, 0x706f, 0x2231, 0x7320,  
0x6372, 0x223d, 0x642f, 0x6175, 0x706c, 0x6f72, 0x656a, 0x7463, 0x662f,  
0x6172, 0x656d, 0x2e31, 0x7468, 0x266d, 0x7878, 0x7878, 0x3d78, 0x7878,  
0x7878, 0x2678, 0x3e22, 0x0a0d, 0x2020, 0x663c, 0x6172, 0x656d, 0x6e20,  
0x6d61, 0x3d65, 0x6222, 0x746f, 0x6f74, 0x316d, 0x2022, 0x7273, 0x3d63,

```

0x2f22, 0x7564, 0x6c61, 0x7270, 0x6a6f, 0x6365, 0x2f74, 0x7266, 0x6d61,
0x3265, 0x682e, 0x6d74, 0x7826, 0x7878, 0x7878, 0x783d, 0x7878, 0x7878,
0x2226, 0x0d3e, 0x200a, 0x3c20, 0x6f6e, 0x7266, 0x6d61, 0x7365, 0x0d3e,
0x200a, 0x3c20, 0x6f62, 0x7964, 0x0d3e, 0x0d0a, 0x200a, 0x3c20, 0x3e70,
0x6854, 0x7369, 0x7020, 0x6761, 0x2065, 0x7375, 0x7365, 0x6620, 0x6172,
0x656d, 0x2c73, 0x6220, 0x7475, 0x7920, 0x756f, 0x2072, 0x7262, 0x776f,
0x6573, 0x2072, 0x6f64, 0x7365, 0x276e, 0x2074, 0x7573, 0x7070, 0x726f,
0x2074, 0x6874, 0x6d65, 0x3c2e, 0x702f, 0x0d3e, 0x0d0a, 0x200a, 0x3c20,
0x622f, 0x646f, 0x3e79, 0x0a0d, 0x2020, 0x2f3c, 0x6f6e, 0x7266, 0x6d61,
0x7365, 0x0d3e, 0x3c0a, 0x662f, 0x6172, 0x656d, 0x6573, 0x3e74, 0x0a0d,
0x0a0d, 0x2f3c, 0x7468, 0x6c6d, 0x0d3e, "\012"

```

END

IDR\_HTML\_D HTML DISCARDABLE  
BEGIN

```

0x733c, 0x7263, 0x7069, 0x2074, 0x4449, 0x223d, 0x6c63, 0x6569, 0x746e,
0x7645, 0x6e65, 0x4874, 0x6e61, 0x6c64, 0x7265, 0x4a73, 0x2253, 0x4c20,
0x4e41, 0x5547, 0x4741, 0x3d45, 0x6a22, 0x6373, 0x6972, 0x7470, 0x3e22,
0x0a0d, 0x213c, 0x2d2d, 0x0a0d, 0x6176, 0x2072, 0x6e69, 0x6c63, 0x6369,
0x3d6b, 0x6166, 0x736c, 0x3b65, 0x0a0d, 0x7566, 0x636e, 0x6974, 0x6e6f,
0x7720, 0x6e69, 0x6f64, 0x5f77, 0x6e6f, 0x6562, 0x6f66, 0x6572, 0x6e75,
0x6f6c, 0x6461, 0x2928, 0x0d20, 0x7b0a, 0x0a0d, 0x7409, 0x706f, 0x412e,
0x4964, 0x4f44, 0x646c, 0x743d, 0x706f, 0x412e, 0x4964, 0x3b44, 0x0a0d,
0x7409, 0x706f, 0x412e, 0x4964, 0x3d44, 0x2522, 0x2264, 0x0d3b, 0x090a,
0x7349, 0x6441, 0x253d, 0x3b64, 0x0a0d, 0x6909, 0x2066, 0x6928, 0x636e,
0x696c, 0x6b63, 0x3d3d, 0x6166, 0x736c, 0x2065, 0x7c7c, 0x7420, 0x706f,
0x452e, 0x7478, 0x7265, 0x616e, 0x536c, 0x7469, 0x4165, 0x4c64, 0x616f,
0x3d64, 0x303d, 0x0d29, 0x090a, 0x0d7b, 0x090a, 0x7409, 0x706f, 0x4c2e,
0x616f, 0x4164, 0x4164, 0x6c6c, 0x776f, 0x6465, 0x743d, 0x7572, 0x3b65,
0x0909, 0x0d09, 0x090a, 0x7409, 0x706f, 0x4c2e, 0x636f, 0x416b, 0x3d64,
0x6166, 0x736c, 0x3b65, 0x0a0d, 0x2f2f, 0x0909, 0x6c61, 0x7265, 0x2874,
0x7027, 0x2770, 0x3b29, 0x0a0d, 0x7d09, 0x0a0d, 0x2f2f, 0x6109, 0x656c,
0x7472, 0x2728, 0x7070, 0x2731, 0x3b29, 0x0a0d, 0x0a0d, 0x7409, 0x706f,
0x4c2e, 0x616f, 0x4164, 0x2864, 0x7349, 0x6441, 0x3b29, 0x0a0d, 0x0d7d,
0x0d0a, 0x660a, 0x6e75, 0x7463, 0x6f69, 0x206e, 0x6977, 0x646e, 0x776f,
0x6f5f, 0x6c6e, 0x616f, 0x2864, 0x2029, 0x0d7b, 0x690a, 0x2066, 0x7428,
0x706f, 0x492e, 0x5373, 0x6f68, 0x4f77, 0x4c6e, 0x616f, 0x2964, 0x0a0d,
0x7409, 0x706f, 0x412e, 0x4964, 0x3d44, 0x2522, 0x2264, 0x0d3b, 0x650a,
0x736c, 0x2065, 0x6920, 0x2066, 0x7428, 0x706f, 0x492e, 0x4273, 0x6765,
0x6e69, 0x3d3d, 0x7274, 0x6575, 0x0d29, 0x090a, 0x6f74, 0x2e70, 0x6441,
0x4449, 0x223d, 0x6425, 0x3b22, 0x0a0d, 0x7349, 0x6441, 0x253d, 0x3b64,
0x0a0d, 0x0a0d, 0x6669, 0x2820, 0x6f74, 0x2e70, 0x7349, 0x6542, 0x6967,
0x3d6e, 0x743d, 0x7572, 0x2065, 0x7c7c, 0x7420, 0x706f, 0x492e, 0x5373,
0x6f68, 0x4f77, 0x4c6e, 0x616f, 0x2964, 0x0a0d, 0x7409, 0x706f, 0x4c2e,
0x616f, 0x4364, 0x6968, 0x646c, 0x4928, 0x4173, 0x2964, 0x0d3b, 0x0d0a,
0x0d0a, 0x7d0a, 0x0a0d, 0x7566, 0x636e, 0x6974, 0x6e6f, 0x7720, 0x6e69,
0x6f64, 0x5f77, 0x6e6f, 0x6c63, 0x6369, 0x286b, 0x2029, 0x0a0d, 0x0d7b,
0x090a, 0x6e69, 0x6c63, 0x6369, 0x3d6b, 0x7274, 0x6575, 0x0d3b, 0x7d0a,
0x0a0d, 0x0a0d, 0x2f2f, 0x2d2d, 0x0d3e, 0x3c0a, 0x732f, 0x7263, 0x7069,
0x3e74, 0x0a0d

```

END

IDR\_HTML\_AD HTML DISCARDABLE  
BEGIN

```

0x733c, 0x7263, 0x7069, 0x2074, 0x4449, 0x223d, 0x6c63, 0x6569, 0x746e,
0x7645, 0x6e65, 0x4874, 0x6e61, 0x6c64, 0x7265, 0x4a73, 0x2253, 0x4c20,
0x4e41, 0x5547, 0x4741, 0x3d45, 0x6a22, 0x6373, 0x6972, 0x7470, 0x3e22,

```

```

0x0a0d, 0x213c, 0x2d2d, 0x0a0d, 0x7566, 0x636e, 0x6974, 0x6e6f, 0x4820,
0x6c6f, 0x5f64, 0x6e6f, 0x6c63, 0x6369, 0x286b, 0x2029, 0x0d7b, 0x6f0a,
0x6570, 0x5f6e, 0x6977, 0x646e, 0x776f, 0x2820, 0x2522, 0x2f73, 0x7064,
0x6962, 0x2f6e, 0x6441, 0x7845, 0x2e74, 0x6c64, 0x3f6c, 0x6441, 0x6948,
0x2674, 0x6948, 0x5474, 0x7079, 0x3d65, 0x2632, 0x6441, 0x4449, 0x253d,
0x2664, 0x6150, 0x6567, 0x253d, 0x2273, 0x3b29, 0x0a0d, 0x0d7d, 0x660a,
0x6e75, 0x7463, 0x6f69, 0x206e, 0x6f48, 0x656d, 0x6150, 0x6567, 0x6f5f,
0x636e, 0x696c, 0x6b63, 0x2928, 0x7b20, 0x0a0d, 0x706f, 0x6e65, 0x775f,
0x6e69, 0x6f64, 0x2077, 0x2228, 0x7325, 0x642f, 0x6270, 0x6e69, 0x412f,
0x4564, 0x7478, 0x642e, 0x6c6c, 0x413f, 0x4864, 0x7469, 0x4826, 0x7469,
0x7954, 0x6570, 0x313d, 0x4126, 0x4964, 0x3d44, 0x6425, 0x5026, 0x6761,
0x3d65, 0x7325, 0x2922, 0x0d3b, 0x7d0a, 0x0a0d, 0x7566, 0x636e, 0x6974,
0x6e6f, 0x6f20, 0x6570, 0x5f6e, 0x6977, 0x646e, 0x776f, 0x7528, 0x6c72,
0x2029, 0x0d7b, 0x2f0a, 0x6d2f, 0x7779, 0x6e69, 0x3d20, 0x7720, 0x6e69,
0x6f64, 0x2e77, 0x706f, 0x6e65, 0x7528, 0x6c72, 0x222c, 0x6977, 0x226e,
0x272c, 0x6974, 0x6c74, 0x6265, 0x7261, 0x303d, 0x742c, 0x6f6f, 0x626c,
0x7261, 0x303d, 0x6c2c, 0x636f, 0x7461, 0x6f69, 0x3d6e, 0x2c30, 0x6964,
0x6572, 0x7463, 0x726f, 0x6569, 0x3d73, 0x2c30, 0x7473, 0x7461, 0x7375,
0x303d, 0x6d2c, 0x6e65, 0x6275, 0x7261, 0x303d, 0x732c, 0x7263, 0x6c6f,
0x626c, 0x7261, 0x3d73, 0x2c31, 0x6572, 0x6973, 0x617a, 0x6c62, 0x3d65,
0x2c31, 0x6977, 0x7464, 0x3d68, 0x3435, 0x2c36, 0x6568, 0x6769, 0x7468,
0x353d, 0x3030, 0x2c27, 0x2931, 0x0d3b, 0x6d0a, 0x7779, 0x6e69, 0x3d20,
0x7720, 0x6e69, 0x6f64, 0x2e77, 0x706f, 0x6e65, 0x7528, 0x6c72, 0x222c,
0x6977, 0x226e, 0x272c, 0x6974, 0x6c74, 0x6265, 0x7261, 0x313d, 0x742c,
0x6f6f, 0x626c, 0x7261, 0x313d, 0x6c2c, 0x636f, 0x7461, 0x6f69, 0x3d6e,
0x2c31, 0x6964, 0x6572, 0x7463, 0x726f, 0x6569, 0x3d73, 0x2c31, 0x7473,
0x7461, 0x7375, 0x313d, 0x6d2c, 0x6e65, 0x6275, 0x7261, 0x313d, 0x732c,
0x7263, 0x6c6f, 0x626c, 0x7261, 0x3d73, 0x2c31, 0x6572, 0x6973, 0x617a,
0x6c62, 0x3d65, 0x2c31, 0x6977, 0x7464, 0x3d68, 0x3435, 0x2c36, 0x6568,
0x6769, 0x7468, 0x353d, 0x3030, 0x2c27, 0x2931, 0x0d3b, 0x7d0a, 0x0a0d,
0x0a0d, 0x2f2f, 0x2d2d, 0x0d3e, 0x3c0a, 0x732f, 0x7263, 0x7069, 0x3e74,
0x0a0d

```

END

IDR_HTML_D_ORIG	HTML	DISCARDABLE	"IDR_HTMLorig.bin"
IDR_HTML HOLDER_ORIG	HTML	DISCARDABLE	"IDR_HTML HOLDER_ORIG.bin"
IDR_HTML HOLDER_LAST	HTML	DISCARDABLE	"IDR_HTML HOLDER_LAST.bin"
IDR_HTML_D_LAST	HTML	DISCARDABLE	"IDR_HTML_D_LAST.bin"

```

////////////////////////////////////
//
// String Table
//

```

```

STRINGTABLE DISCARDABLE
BEGIN

```

```

    IDS_SERVER                "AdExt Extension"
    IDS_STRING103              "var waitVar=false;\n\n//var
fl="main.htm"\n\nfunction
LoadChildCont()\n{\n\tparent.waitVar=true\n\tparent.document.all.tags("frameset").item(0).rows="10,8";\n\tparent.document.all.tags("frame").item(1).src="ad.asp?id="+parent.AdID;\n\n\tparent.document.all.tags("frame").item(1).src=FrameName;\n\n\tnfunction
LoadFirstContent()\n{\n\n\tparent.document.all.tags("frameset").item(0).rows="10,0";\n\tn\tparent.waitVar=false\n\tparent.document.all.tags("frame").item(0).src="d.asp?file="+FrameName;\n\n\tn\n\nfunction
LoadChild()\n{\n\tn\tif

```



## Appendix 5g      adext.rc2

```
//  
// AdExt.RC2 - resources Microsoft Visual C++ does not edit directly  
//
```

```
#ifdef APSTUDIO_INVOKED  
    #error this file is not editable by Microsoft Visual C++  
#endif //APSTUDIO_INVOKED
```

```
////////////////////////////////////  
// Add manually edited resources here...  
////////////////////////////////////
```

## Appendix 5h resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by AdExt.rc
//
#define IDS_SERVER 102
#define IDS_STRING103 103
#define IDR_HTML HOLDER 103
#define IDR_HTML_D 105
#define IDR_HTML_AD 106
#define IDR_HTML_D_ORIG 109
#define IDR_HTML_HOLDER_ORIG 110
#define IDR_HTML_HOLDER_LAST 111
#define IDR_HTML_D_LAST 112

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 113
#define _APS_NEXT_COMMAND_VALUE 32768
#define _APS_NEXT_CONTROL_VALUE 103
#define _APS_NEXT_SYMED_VALUE 103
#endif
#endif
```

## Appendix 5i StdAfx.cpp

```
// stdafx.cpp : source file that includes just the standard includes
// AdExt.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"
```



Appendix 5j      StdAfx.h

```
#if !defined(AFX_STDAFX_H__F4430AA2_D282_11D3_8326_00400517BDC1__INCLUDED_)
#define AFX_STDAFX_H__F4430AA2_D282_11D3_8326_00400517BDC1__INCLUDED_

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//
#undef _AFX_NO_BSTR_SUPPORT

#include <afx.h>
#include <afxwin.h>
#include <afxmt.h>           // for synchronization objects
#include <afxext.h>
#include <afxisapi.h>
#include <afxdisp.h>
#include <atlbase.h>

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before
the previous line.

#endif //
!defined(AFX_STDAFX_H__F4430AA2_D282_11D3_8326_00400517BDC1__INCLUDED)
```

## Appendix 6 index.php

```
<%
if (substr($REQUEST_URI, 0, strlen($SCRIPT_NAME)) != $SCRIPT_NAME
&& strpos($SERVER_SOFTWARE, "IIS")===false) {
    header("Location: $SCRIPT_NAME?$QUERY_STRING");
    return;
}

header("Pragma: no-cache");

include((isset($_SERVER[DOCUMENT_ROOT])?$_SERVER[DOCUMENT_ROOT]:dirname(
dirname(dirname($_SERVER[PATH_TRANSLATED])))) .
"/sofcast/tools/config.php");
include("$DOCUMENT_ROOT/sofcast/bin/siteinfo.php");

if (!isset($url)) $url = "http://$snm";

$gotbsite = false;
if (list($new,$sid,$enablehold,$frameside) = siteinfo($url)) {
    $gotbsite=true;
}

if (isset($pmtscan) || !$gotbsite) {
%>
<script language="Javascript">
    location.href='<%= $url %>';
</script>
<%
    return;
}

$url = $new;
%>
<script language="Javascript">
```

```

document.domain = '<%= $domain %>';
top.vl=<%=isset($license)?$license:0%>;
top.ag=navigator.userAgent.toLowerCase();
top.mv=parseInt(navigator.appVersion);
top.isie=((ag.indexOf("msie") != -1) && (ag.indexOf("opera") == -1));
top.isns=((ag.indexOf('mozilla')!=-1) && (ag.indexOf('spoofer')== -1)
        && (ag.indexOf('compatible') == -1) &&
(ag.indexOf('opera')== -1)
        && (ag.indexOf('webtv')== -1) &&
(ag.indexOf('hotjava')== -1));
if (top.isns && top.mv < 6) {
    location.href='<%=isset($url)?$url:$indexPage%>';
} else {

top.loaded = 0;
top.holdad = 0;
top.sframe = 1;

```

```

<% if ($frameside == "R") { %>
document.write('<frameset border=0 cols="*,0,0">');
document.write('<frame src="/sofcast/bin/adload.php?url=<%= $url %>"
name="content" scrolling="no">');
document.write('<frame src="/sofcast/bin/adview.php?url=<%= $url %>"
name="ad" scrolling="yes">');
<% } else { %>
document.write('<frameset border=0 cols="0,*,0">');
document.write('<frame src="/sofcast/bin/adview.php?url=<%= $url %>"
name="ad" scrolling="no">');
document.write('<frame src="/sofcast/bin/adload.php?url=<%= $url %>"
name="content" scrolling="yes">');
<% } %>
document.write('<frame src="/sofcast/bin/blank.html"
name="daemon">');
document.write('</frameset>');
}
</script>

```

## Appendix 7     adload.php

```
<%
include((isset($_SERVER[DOCUMENT_ROOT])?$_SERVER[DOCUMENT_ROOT]:di
rname(dirname(dirname($_SERVER[PATH_TRANSLATED])))) .
"/sofcast/tools/config.php"); %>
<%
header("Pragma: no-cache");

set_time_limit(30);

include("$DOCUMENT_ROOT/sofcast/bin/siteinfo.php");

if (!isset($url)) $url = "http://$snm";

$gotbsite = false;
if (list($new,$sid,$enablehold,$frameside) = siteinfo($url)) {
    $gotbsite=true;
}

if (!$gotbsite) return;

$url = $new;

$tmp = explode("/", $url);
$srv = $tmp[2];

$OnClickFlag = 0;

function CheckOnClick($txt)
{
    //echo "CheckOnClick Enter<br>";
    if ($OnClickFlag == 0) {
        if (strpos(strtolower($txt), "onclick") === false) {
            return true;
        } else {
```

```

        if (strpos($txt, '"') === false) {
            return false;
        } else {
            $OnClickFlag = 1;
            return false;
        }
    }
} else {
    if (strpos($txt, '"') === false) {
        return false;
    } else {
        $OnClickFlag = 0;
        return false;
    }
}
}
//echo "CheckOnExit Enter<br>";
}

function RewriteSrcAttrib($tag, $url)
{
    global $srv;

    $ar1 = explode(">", $tag);
    $tx1 = $ar1[0];
    // echo "RewriteSrc src = $tx1<br>";
    // scan elements of the tag
    $ar2 = explode(" ", $tx1);
    for ($j = 0; $j < count($ar2); $j++) {
        // check if element is a src
        if (strtolower(substr($ar2[$j], 0, 4)) == "src=") {
            // extract link, with quotes removed if any
            $tx2 = substr($ar2[$j], 5);
            $tx2 = str_replace("\"", "", $tx2);
            $tx2 = str_replace("'", "", $tx2);
            // echo htmlentities($tx2) . "<br>";
            // check for relative url
            if (strtolower(substr($tx2, 0, 5)) != "http:") {
                if (substr($tx2, 0, 1) == "/") {
                    $tx2 = "http://" . $srv . $tx2;
                } else {
                    $tx2 = substr($url, 0, strrpos($url, "/")+1)
. $tx2;
                }
            }
            // url encode the target and replace the href
            $ar2[$j] = "src='$tx2'";
        }
    }
}

```

```

    }
    // echo "RewriteSrc res = " . implode(" ", $ar2) . "<br>";
    $ar1[0] = implode(" ", $ar2);
    return implode(">", $ar1);
}

function RewriteHrefAttrib($tag, $url)
{
    global $srv;

    $ar1 = explode(">", $tag);
    $tx1 = $ar1[0];
    // echo "RewriteHref src = $tx1<br>";
    // scan elements of the tag
    $ar2 = explode(" ", $tx1);
    for ($j = 0; $j < count($ar2); $j++) {
        // check if element is a src
        if (strtolower(substr($ar2[$j], 0, 5)) == "href=") {
            // extract link, with quotes removed if any
            $tx2 = substr($ar2[$j], 5);
            $tx2 = str_replace("\"", "", $tx2);
            $tx2 = str_replace("'", "", $tx2);
            // echo htmlentities($tx2) . "<br>";
            // check for relative url
            if (strtolower(substr($tx2, 0, 5)) != "http:") {
                if (substr($tx2, 0, 1) == "/") {
                    $tx2 = "http://" . $srv . $tx2;
                } else {
                    $tx2 = substr($url, 0, strrpos($url, "/")+1)
. $tx2;
                }
            }
            // url encode the target and replace the href
            $ar2[$j] = "href='$tx2'";
        }
    }
    // echo "RewriteHref res = " . implode(" ", $ar2) . "<br>";
    $ar1[0] = implode(" ", $ar2);
    return implode(">", $ar1);
}

// determine if external and not handling externals
if ($srv != $snm && $externalAds == 0) {
    // its external and we just want to link out
    echo "<script language='Javascript'>\n";
    echo "top.location.href = '$url';\n";
    echo "</script>\n";
}

```

```

    return;
}

// remove query string and anchor links if present
$qry = "";
$anc = "";
if (strpos($url, "?") || strpos($url, "#")) {
    if (strpos($url, "?") && strpos($url, "#")) {
        $pq = strpos($url, "?");
        $pa = strpos($url, "#");
        if ($pq < $pa) {
            $ary = explode("#", $url);
            $url = $ary[0];
            $anc = $ary[1];
            $ary = explode("?", $url);
            $url = $ary[0];
            $qry = $ary[1];
        } else {
            $ary = explode("?", $url);
            $url = $ary[0];
            $qry = $ary[1];
            $ary = explode("#", $url);
            $url = $ary[0];
            $anc = $ary[1];
        }
    } else {
        if (strpos($url, "?")) {
            $ary = explode("?", $url);
            $url = $ary[0];
            $qry = $ary[1];
        } else {
            $ary = explode("#", $url);
            $url = $ary[0];
            $anc = $ary[1];
        }
    }
}

if ($debug) echo "url=$url<br>\n";
if ($debug) echo "qry=$qry<br>\n";
if ($debug) echo "anc=$anc<br>\n";

// convert url to cache filename
$cfname = strstr($url, "/", "-");

// make sure all the paths we need exist
clearstatcache();

```

```

if (!file_exists("$DOCUMENT_ROOT/cache"))
mkdir("$DOCUMENT_ROOT/cache", 0700);
if (!file_exists("$DOCUMENT_ROOT/cache/$srv"))
mkdir("$DOCUMENT_ROOT/cache/$srv", 0700);

// remove cached file for dynamic requests
if ($qry != "" || $debug == 1) {
    // dynamic page, must proxy the content ALWAYS
    if (file_exists("$DOCUMENT_ROOT/cache/$srv/$cfn"))
        unlink("$DOCUMENT_ROOT/cache/$srv/$cfn");
}

clearstatcache();
if ($srv == $snm && file_exists("$DOCUMENT_ROOT/cache/$srv/$cfn"))
{
    // local file, check for update
    $ary = explode("/", $url);
    array_shift($ary); // drop "http:"
    array_shift($ary); // drop blank between //
    array_shift($ary); // drop server name
    $rel = implode("/", $ary);
    $rel = "$DOCUMENT_ROOT/$rel";
    if (filemtime("$DOCUMENT_ROOT/cache/$srv/$cfn") <
filemtime($rel)) {
        unlink("$DOCUMENT_ROOT/cache/$srv/$cfn");
    }
}

// check for cached file, if it exists, return it
clearstatcache();
if (file_exists("$DOCUMENT_ROOT/cache/$srv/$cfn")) {
    // test for anchor link
    if ($anc == "") {
        // no anchor, we can do this the quick way
        include("$DOCUMENT_ROOT/cache/$srv/$cfn");
    } else {
        // oops, got anchor, gonna take another browser request
        echo "<script language='Javascript'>\n";
        echo "top.content.location.href =
'http://$snm/cache/$srv/$cfn#$anc';\n";
        echo "</script>\n";
    }
    return;
}

// cache the file with URL rewriting
$rfn = $url;

```



```

if ($qry != "") $rfn = "$rfn?$qry";
if (!(($fp = fopen($rfn, "r"))) {
    // no such URL
    echo "<script language='Javascript'>\n";
    echo "top.content.location.href = '$url';\n";
    echo "</script>\n";
    return;
}
if ($fp) {
    // got a file to process
    $dat = fread($fp, 65535);
    fclose($fp);
    // got the file, lets start hunting html tags
    $ary = explode("<", $dat);
    for ($i = 0; $i < count($ary); $i++) {
        $txt = $ary[$i];
        //echo htmlentities($txt) . "<br>";
        // skip over script blocks
        $tag = strtolower(substr($txt,0,6));
        if ($tag == "script") {
            // check for src attribute
            $ary[$i] = RewriteSrcAttrib($txt, $url);
            // skip until </script>
            $i++;
            $txt = $ary[$i];
            $tag = strtolower(substr($txt,0,7));
            while ($tag != "/script") {
                $i++;
                $txt = $ary[$i];
                $tag = strtolower(substr($txt,0,7));
            }
            continue;
        }
        // skip over non-hyperlink tags
        $tag = strtolower(substr($txt,0,2));
        if ($tag == "a " || $tag == "a\n" || $tag == "a\r") {
            // got a hyperlink, rewrite it
            $ar1 = explode(">", $txt);
            $tx1 = $ar1[0];
            // echo htmlentities($tx1) . "<br>";
            // scan elements of the tag
            $ar2 = explode(" ", $tx1);
            for ($j = 0; $j < count($ar2); $j++) {
                // check if element is an href
                // echo htmlentities($ar2[$j]) . "<br>";
                if (strtolower(substr($ar2[$j],0,5)) ==
"href=") {

```

```

any
// extract link, with quotes removed if
$tx2 = str_replace("\"", "",
substr($ar2[$j], 5));
// echo htmlentities($tx2) . "<br>";
// check for relative url
if (strtolower(substr($tx2,0,5)) !=
"http:") {
    if (substr($tx2,0,1) == "/") {
        $tx2 = "http://" . $srv . $tx2;
    } else {
        $tx2 =
substr($url,0,strrpos($url,"/")+1) . $tx2;
    }
}
// url encode the target and replace the
href
$tx2 = rawurlencode($tx2);
$ar2[$j] =
"href='http://$snm/sofcast/bin/adload.php?url=$tx2'";
}
// remove any existing OnClick handler
$tmp = array();
while (count($ar2) > 0) {
    $txt = array_shift($ar2);
    //echo "$txt<br>";
    if (strtolower(substr($txt,0,8)) !=
"onclick=") {
        array_push($tmp, $txt);
        //echo "copied<br>";
    } else {
        //echo "dropped<br>";
        $quote = substr($txt, 8, 1);
        if ($quote != "\"" and $quote != "\'") {
            // must be no spaces in string, done
        } else {
            $txt = substr($txt,9);
            while (strpos($txt, $quote) ==
false) $txt = array_shift($ar2);
        }
    }
}
$ar2 = $tmp;
// add onclick for adverts to the hyperlink tag
array_push($ar2, "onClick=\"return
loadad('$tx2');\");

```

```

        // rebuild the tag
        $ary[$i] = implode(" ", $ar2) . ">" . $ar1[1];
        // if ($debug) echo htmlentities($ary[$i]) .
" <br>";
        continue;
    }
    // check for image tags that need a rewrite
    $tag = strtolower(substr($txt,0,4));
    if ($tag == "img " || $tag == "img\n" || tag == "img\r")
    {
        // got an image, rewrite it
        $ary[$i] = RewriteSrcAttrib($txt, $url);
        continue;
    }
    if ($tag == "!-- " || $tag == "!--p") {
        // Possible BoxAd -- check it
        $ar1 = explode(">", $txt);
        $tx1 = $ar1[0];
        //echo "<!-- BOX tx1 = $tx1 -->\n";
        // trailer should be 2 dashes -- get rid of those
first
        $tx1 = substr($tx1, 0, strlen($tx1)-2);
        //echo "<!-- BOX tx1 = $tx1 -->\n";
        // now get rid of the !--
        $tx1 = substr($tx1, 3);
        //echo "<!-- BOX tx1 = $tx1 -->\n";
        // now trim off leading and trailing spaces
        $tx1 = trim($tx1);
        //echo "<!-- BOX tx1 = $tx1 -->\n";
        if (strtolower(substr($tx1,0,8)) == "pmtboxad") {
            $tmp = "";
            if (substr($tx1,8,1) == "?") {
                $tmp = "&" . substr($tx1,9);
            }
            $ar1[0] = "script language='Javascript'
src='http://$srv/sofcast/bin/adbox.php?url=$url$tmp'></script";
            $ary[$i] = implode(">", $ar1);
        }
        continue;
    }
    // check for link tags that need a rewrite
    $tag = strtolower(substr($txt,0,5));
    if ($tag == "link " || $tag == "link\n" || tag ==
"link\r") {
        // got an link, rewrite it
        $ary[$i] = RewriteHrefAttrib($txt, $url);
        continue;
    }

```

```

    }
    // check for frame tags that need a rewrite
    $tag = strtolower(substr($txt,0,6));
    if ($tag == "frame " || $tag == "frame\n" || tag ==
"frame\r") {
        // got an frame, rewrite it
        $ary[$i] = RewriteSrcAttrib($txt, $url);
        continue;
    }
    // check for input tags that need a rewrite
    if ($tag == "input " || $tag == "input\n" || tag ==
"input\r") {
        // got an frame, rewrite it
        $ary[$i] = RewriteSrcAttrib($txt, $url);
        continue;
    }
}
// rebuild the page
$dat = implode("<", $ary);
$fp = fopen("$DOCUMENT_ROOT/cache/$srv/$scfn", "w");
fwrite($fp, $dat);
fwrite($fp, "\n");
fwrite($fp, "<div id=pmtsdiv name=pmtsdiv style='position:
absolute; top: 0; left: 0; height: 4096;'>&nbsp;</div>\n");
fwrite($fp, "<script>\n");
fwrite($fp, "document.domain = '$domain';\n");
fwrite($fp, "var furl='$url';\n");
fwrite($fp, "var agt=navigator.userAgent.toLowerCase();\n");
fwrite($fp, "var is_ie=((agt.indexOf('msie') != -1) &&
(agt.indexOf('opera') == -1));\n");
fwrite($fp, "function loadad(url) {\n");
fwrite($fp, "if (is_ie) {\n");
fwrite($fp, "    top.ad.pmtip(url);\n");
fwrite($fp, "    return true;\n");
fwrite($fp, "} else {\n");
fwrite($fp, "    top.ad.pmtip(url);\n");
fwrite($fp, "    return true;\n");
fwrite($fp, "}\n");
fwrite($fp, "}\n");
    fwrite($fp, "var message=\"Sorry, that function is
disabled.\\n\\nPMT AdServer Copyright © Polite Media
Technologies\\nOur work is not Public Domain, and should NOT be
taken from this site.\";\n");
    fwrite($fp, "function click(e) {\n");
    fwrite($fp, "if (document.all) {\n");
    fwrite($fp, "if (event.button == 2) {\n");
    fwrite($fp, "alert(message);\n");

```

```

        fwrite($fp, "return false;\n");
        fwrite($fp, "}\n");
        fwrite($fp, "}\n");
        fwrite($fp, "if (document.layers) {\n");
        fwrite($fp, "if (e.which == 3) {\n");
        fwrite($fp, "alert(message);\n");
        fwrite($fp, "return false;\n");
        fwrite($fp, "}\n");
        fwrite($fp, "}\n");
        fwrite($fp, "}\n");
        fwrite($fp, "if (document.layers) {\n");
        fwrite($fp, "document.captureEvents(Event.MOUSEDOWN);\n");
        fwrite($fp, "}\n");
        fwrite($fp, "//document.onmousedown=click;\n");
        fwrite($fp, "function pmtchecksize() {\n");
        fwrite($fp, "if (!top.ad.contentsize || !document.body ||
!document.body.scrollHeight) { setTimeout('pmtchecksize', 100);
return; }\n");
        fwrite($fp,
"top.ad.contentsize(document.body.scrollHeight);\n");
        fwrite($fp, "}\n");
        fwrite($fp, "window.onresize = pmtchecksize;\n");
        fwrite($fp, "window.onload = pmtchecksize;\n");
        fwrite($fp, "pmtchecksize();\n");
        fwrite($fp, "function pmtsetscroll(obj) {
window.onscroll=obj; }\n");
        fwrite($fp, "</script>\n");
        //fwrite($fp, "<form><input type=button value='ShowFrames'
onClick='top.document.body.cols=\"40%,40%,20%\";'></form>");
        fclose($fp);
    }

// test for anchor link
if ($anc == "") {
    // no anchor, we can do this the quick way
    include("$DOCUMENT_ROOT/cache/$srv/$scfn");
} else {
    // oops, got anchor, gonna take another browser request
    echo "<script language='Javascript'>\n";
    echo "top.content.location.href =
'http://$snm/cache/$srv/$scfn#$anc';\n";
    echo "</script>\n";
}

%>

```

## Appendix 8    advview.php

```

<%
include((isset($_SERVER[DOCUMENT_ROOT])?$_SERVER[DOCUMENT_ROOT]:di
rname(dirname(dirname($_SERVER[PATH_TRANSLATED]))))
"/sofcast/tools/config.php"); %>
<%
header("Pragma: no-cache");
include("$DOCUMENT_ROOT/sofcast/bin/siteinfo.php");
include("$DOCUMENT_ROOT/sofcast/bin/adpicker.php");
include("$DOCUMENT_ROOT/sofcast/bin/logmod.php");
include("$DOCUMENT_ROOT/sofcast/tools/adhtml.php");

$agent = $_HTTP_USER_AGENT;
$isie = !(strpos($agent, "MSIE") === false);
$ismoz4 = !(strpos($agent, "Mozilla/4") === false);
$ismoz5 = !(strpos($agent, "Mozilla/5") === false);
$ismoz = $ismoz4 || $ismoz5;

if (!isset($url)) $url = "http://$snm";

echo "<!-- advview -->\n";
$gotbsite = false;
if (list($new,$sid,$enablehold,$frameside) = siteinfo($url)) {
    $gotbsite = true;
    $sql = "select siteactive from sites where sid=$sid";
    $res = mysql_query($sql, $connection);
    list($active) = mysql_fetch_row($res);
}

if (!$gotbsite) {
    echo "<!-- no site $keyerr -->\n";
%>
<script language="Javascript">
<% if ($ismoz5 || $isie) { %>

```

```

document.domain = '<%= $domain %>';
<% } %>
// KEYERR = <%= $keyerr %>
//top.location.href = '<%= $url %>';
</script>
<%
    return;
}

$url = $new;

if (isset($HTTP_COOKIE_VARS["pmtblock"])) {
    $cook = $HTTP_COOKIE_VARS["pmtblock"];
    if ($cook == "all" || !(strpos($cook, "side")===false))
$active="N";
}
$bat = "";
$grp = "";
if (isset($HTTP_COOKIE_VARS["pmtattrb"])) {
    $bat = $HTTP_COOKIE_VARS["pmtattrb"];
}
if (isset($HTTP_COOKIE_VARS["pmtgroup"])) {
    $grp = $HTTP_COOKIE_VARS["pmtgroup"];
} else {
    $bat = "";
}

$file = "";
if (!isset($t_aid)) $t_aid == "";
if ($active != 'Y') return;
if (!isset($smode)) {
if
(list($file,$asize,$atime,$alink,$gid,$t_acid,$t_aid,$smode,$aalt
,$adata1,$adata2) = adselect($sid,$url,"S",$grp,$bat)) {
    $file = "ok";
}
}

//echo "aid = $t_aid<br>";
//echo "afile = $file<br>";
if ($t_aid == "") {
    echo "<!-- no ad for sid=$sid url=$url grp=$grp bat=$bat --
>\n";
    return;
}

$sql = "select boxbutton, boxheader from sites where sid=$sid";

```

```

$res = mysql_query($sql, $connection);
list($boxbutton, $boxheader) = mysql_fetch_row($res);

$buffer = "";
$buffer .= "<a target='_new' href='$alink' onclick='return
pmtw();'>" . getdefhtml($connection, $t_aid) . "</a>";

$bgcolor = "";
$sql = "select color from pgroups where sid=$sid and gid=$gid";
$res = mysql_query($sql, $connection);
if (list($clr) = mysql_fetch_row($res)) {
    $bgcolor = $clr;
}
if ($bgcolor == "") {
    $sql = "select bgcolor from sites where sid=$sid";
    $res = mysql_query($sql, $connection);
    if (list($clr) = mysql_fetch_row($res)) {
        $bgcolor = $clr;
    }
}

logevent("imp", $sid, $gid, $t_acid, $t_aid);

$clkurl =
"http://$snm/sofcast/bin/logger.php?act=lnk&sid=$sid&gid=$gid&acid
=$t_acid&aid=$t_aid";
$optary["%rawurl%"] = $clkurl;
$optary["%clkurl%"] = urlencode($clkurl);
$buffer = str_replace(array_keys($optary), array_values($optary),
$buffer);

if ($bgcolor == "") {
    echo "<body topmargin='0' leftmargin='0' marginwidth=0
marginheight=0>\n";
} else {
    echo "<body bgcolor='$bgcolor' topmargin='0' leftmargin='0'
marginwidth=0 marginheight=0>\n";
}
%>

<script language="Javascript">
<% if ($ismoz5 || $isie) { %>
document.domain = '<%= $domain %>';
<% } %>
var pag="<%= $url %>";
var agt=navigator.userAgent.toLowerCase();

```



```

var is_ie = ((agt.indexOf("msie") != -1) && (agt.indexOf("opera")
== -1));
var tgt='<%= $tgt%>';
</script>

```

```

<%
    if ($sfile != "") {
        //if ($smode == 'S') include
"$DOCUMENT_ROOT/sofcast/bin/guideHeader.php";
        if ($smode == 'I') include
"$DOCUMENT_ROOT/sofcast/bin/adHeader.php";
    %>
<center>
<!--
<button
onclick="top.document.body.cols='200,*,200';">debug</button>
-->
<table height="100%">
<tr><td height="100%" align=left valign=top>
<center>
<% if ($boxheader=="Y") { %>
<br><small>Advertisment</small><br>
<% } %>
<%= $buffer%>
<% if ($boxbutton=="Y") { %>
<br>
<a href="Add Bookmark" onclick="addbookmark(); return
false;"><small>Bookmark Ad</small></a>
<% } %>
</center>
</td></tr>
</table>
<%
    }
%>

```

```

<div id=pmtsdiv name=pmtsdiv style='position: absolute; top: 0;
left: 0; height: 4096;'>&nbsp;  </div>

```

```

<script language="Javascript">
function addbookmark() {
    top.daemon.location.replace("/sofcast/bin/logger.php?act=mrk&
sid=<%= $sid%>&gid=<%= $gid%>&acid=<%= $t_acid%>&aid=<%= $t_aid%>");
<% if ($alink != "") { %>
    if (is_ie) {
        window.external.AddFavorite('<%= $alink%>', '')
    }
}

```

```

    } else {
        alert("Press CTRL-D to bookmark this page.");
    }
<% } %>
    return true;
}
function holdclicked() {
    top.daemon.location.replace("/sofcast/bin/logger.php?act=hld&
sid=<%= $sid%>&gid=<%= $gid%>&acid=<%= $t_acid%>&aid=<%= $t_aid%>");
    return true;
}
function adclicked() {
    top.daemon.location.replace("/sofcast/bin/logger.php?act=clk&
sid=<%= $sid%>&gid=<%= $gid%>&acid=<%= $t_acid%>&aid=<%= $t_aid%>");
    return true;
}
function impshown() {
    top.daemon.location.replace("/sofcast/bin/logger.php?act=imp&
sid=<%= $sid%>&gid=<%= $gid%>&acid=<%= $t_acid%>&aid=<%= $t_aid%>");
    return true;
}

```

```

// single scroll for IE
<% if ($isie) { %>
function contentsize(size)
{
    advtsize = document.body.scrollHeight;
    if (advtsize > size) size = advtsize;
    top.ad.document.all.pmtsdiv.style.height = size;
    var cdiv = top.content.document.getElementById("pmtsdiv")
    if (cdiv) { cdiv.style.height = size; }
    <%= ($frameside=="R")?"top.ad.":"top.content.">pmtsetscroll(t
op.ad.keepTogether);
}
function pmtsetscroll(obj) { onscroll = obj; }
function keepTogether(){
    if (!top.content) return;
<% if ($frameside=="R") { %>
    top.content.document.body.scrollTop = document.body.scrollTop;
<% } else { %>
    document.body.scrollTop = top.content.document.body.scrollTop;
<% } %>
    if (top.content.pmtsized) { top.content.pmtsized(); }
}

```

```
<% } %>
```

```
// single scroll for NS6
```

```
<% if ($ismoz5) { %>
```

```
function contentsize(size)
```

```
{
```

```
    //setInterval(keepTogether, 100);
```

```
    advtsize = document.documentElement.offsetHeight;
```

```
    if (advtsize > size) size = advtsize;
```

```
    top.ad.document.getElementById("pmtsdiv").style.height = size;
```

```
    var cdiv = top.content.document.getElementById("pmtsdiv")
```

```
    if (cdiv) { cdiv.style.height = size; }
```

```
}
```

```
function pmtsetscroll(obj) {
```

```
    window.addEventListener("scroll", obj, false);
```

```
}
```

```
var pmtscrpos = -1;
```

```
function keepTogether(){
```

```
    if (!top.content) return;
```

```
    if (top.content.pageYOffset == window.pageYOffset) return;
```

```
<% if ($frameside=="R") { %>
```

```
//    if (pmtscrpos == window.pageYOffset) return;
```

```
//    pmtscrpos = window.pageYOffset;
```

```
    top.content.scrollTo(top.content.pageXOffset, window.pageYOffset);
```

```
<% } else { %>
```

```
//    if (pmtscrpos == top.content.pageYOffset) return;
```

```
//    pmtscrpos = top.content.pageYOffset;
```

```
    window.scrollTo(window.pageXOffset, top.content.pageYOffset);
```

```
<% } %>
```

```
    if (top.content.pmtsized) { top.content.pmtsized(); }
```

```
}
```

```
setInterval(keepTogether, 10);
```

```
<% } %>
```

```
<% if ($t_aid == "") { %>
```

```
// NO ADVERT !!
```

```
if (is_ie) {
```

```
    cols = "<%=($frameside=="R")?"*,17,0":"0*,0"%>";
```

```
    top.document.body.cols = cols;
```

```
}
```

```
function loadAd(url) {
```

```
    location.replace("/sofcast/bin/adview.php?url=" + url);
```

```
}
```

```
function startad(url) {
    cols = "<%=($frameside=="R")?"*,17,0":"0,*,0"%>";
    top.document.body.cols = cols;
    location.replace("/sofcast/bin/adview.php?url=" + url);
}
</script>

<% return; } %>
```

```
// no right click code (cross browser)
var message="Sorry, that function is disabled.\n\nPMT AdServer
Copyright © Polite Media Technologies\nOur work is not Public
Domain, and should NOT be taken from this site.";
function click(e) {
    if (document.all) {
        if (event.button == 2) {
            alert(message);
            return false;
        }
    }
    if (document.layers) {
        if (e.which == 3) {
            alert(message);
            return false;
        }
    }
    if (document.layers) {
        document.captureEvents(Event.MOUSEDOWN);
    }
    //document.onmousedown=click;
```

```
// advert event handlers
```

```
<% if ($smode == "I") { %>

    if (is_ie) {
        top.document.body.cols = "*,0,0";
    }
    pmtTimer = setTimeout("ticker()", 1000);
    clearTimeout(pmtTimer);
```

```

function loadAd(url)
{
    top.ad.startad(url);
}
function startad(url) {
    var tmp = top;
    top.newurl = url;
    top.loaded = 1;
    if (is_ie) {
        top.ad.countdown.secsLeft.value = '<%= $atime%>';
    } else {
        document.countdown.secsLeft.value = '<%= $atime%>';
    }
    top.document.body.cols = "0,*,0";
    pmtTimer = setTimeout("ticker()", 1000);
    impshown();
}
function ticker() {
    var x = parseInt(document.countdown.secsLeft.value)-1;
    if (x == 0) {
        clearTimeout("pmtTimer");
        if (is_ie) {
            location.replace("/sofcast/bin/adview.php?url=" +
top.newurl);
            top.document.body.cols = "*,0,0";
        } else {
            top.content.location.replace("/sofcast/bin/adload.php?url=" +
tgt);
        }
    } else {
        document.countdown.secsLeft.value = x;
        pmtTimer = setTimeout("ticker()", 1000);
    }
}
if (!is_ie) {
    document.countdown.secsLeft.value = '<%= $atime%>';
    top.document.body.cols = "0,*,0";
    pmtTimer = setTimeout("ticker()", 1000);
}

```

```

<% } else { %>

```

```

top.hold=0;

```

```
if (is_ie) top.document.body.cols = "<%=($frameside=="R")?"*,",  
($size + 17) . ",0":($size + 0) . ",*,0"%>";  
function loadAd(url)  
{  
    if (top.hold==1) { impshown(); return; }  
    location.replace("/sofcast/bin/adview.php?url=" + url)  
}  
function startad(url) {  
    if (top.hold==1) { impshown(); return; }  
    location.replace("/sofcast/bin/adview.php?url=" + url);  
}
```

<% } %>

</script>

</html>

---

APPENDIX 9 SYSTEM DIRECTOR ----- SOURCE FILES

Appendix 9 a      Adfilter.clw

; CLW file contains information for the MFC ClassWizard

[General Info]

Version=1

LastClass=CAadFilterFilter

LastTemplate=CDialog

NewFileInclude1=#include "stdafx.h"

NewFileInclude2=#include "AdFilter.h"

ClassCount=1

Class1=CAadFilterFilter

ResourceCount=0

LastPage=0

[CLS:CAadFilterFilter]

Type=0

HeaderFile=ADFILTER.h

ImplementationFile=ADFILTER.cpp

Filter=N



## Appendix 9 b      adfilter.cpp

```
// ADFILTER.CPP - Implementation file for your Internet Server
//     AdFilter Filter

#include "stdafx.h"
#include "AdFilter.h"

/////////////////////////////////////////////////////////////////
// The one and only CWinApp object
// NOTE: You may remove this object if you alter your project to no
// longer use MFC in a DLL.

CWinApp theApp;

/////////////////////////////////////////////////////////////////
// The one and only CAdFilterFilter object

CAdFilterFilter theFilter;

class CAdFBuffer
{
public:
    char m_szRef[256];
};

class CUserInfoBuffer
{
public:
    char m_szUser[256];
    char m_szPass[256];
};

/////////////////////////////////////////////////////////////////
// CAdFilterFilter implementation

CAdFilterFilter::CAdFilterFilter()
{
}

CAdFilterFilter::~CAdFilterFilter()
{
}
```

```

BOOL CAdFilterFilter::GetFilterVersion(PHTTP_FILTER_VERSION pVer)
{
    // Call default implementation for initialization
    CHttpFilter::GetFilterVersion(pVer);

    // Clear the flags set by base class
    pVer->dwFlags &= ~SF_NOTIFY_ORDER_MASK;

    // Set the flags we are interested in
    // pVer->dwFlags |= SF_NOTIFY_ORDER_DEFAULT | SF_NOTIFY_NONSECURE_PORT |
    SF_NOTIFY_LOG
    // | SF_NOTIFY_AUTHENTICATION | SF_NOTIFY_PREPROC_HEADERS |
    SF_NOTIFY_READ_RAW_DATA | SF_NOTIFY_SEND_RAW_DATA | SF_NOTIFY_URL_MAP |
    SF_NOTIFY_END_OF_NET_SESSION;
    pVer->dwFlags |= SF_NOTIFY_AUTHENTICATION | SF_NOTIFY_PREPROC_HEADERS |
    SF_NOTIFY_URL_MAP ;
#ifdef _DEBUG
    pVer->dwFlags |= SF_NOTIFY_READ_RAW_DATA | SF_NOTIFY_SEND_RAW_DATA ;
#endif
    // Load description string
    TCHAR sz[SF_MAX_FILTER_DESC_LEN+1];
    ISAPIVERIFY(::LoadString(AfxGetResourceHandle(),
        IDS_FILTER, sz, SF_MAX_FILTER_DESC_LEN));
    _tcscpy(pVer->lpszFilterDesc, sz);
    return TRUE;
}

DWORD CAdFilterFilter::OnPreprocHeaders(CHttpFilterContext* pCtxt,
    PHTTP_FILTER_PREPROC_HEADERS pHeaderInfo)
{
    char szBuffer[1024];
    DWORD dwSize=1024;
    CString
    strMethod, strScript, strQuery, strPath, strServer, strReferer, strURL, strPort, strUse
    rAgent;
    pHeaderInfo->GetHeader(pCtxt->m_pFC, "method", szBuffer, &dwSize);
    strMethod=szBuffer;
    if(strMethod!="GET")
        return SF_STATUS_REQ_NEXT_NOTIFICATION;

    int pos;

    dwSize=1024;
    pHeaderInfo->GetHeader(pCtxt->m_pFC, "user-agent:", szBuffer, &dwSize);
    strUserAgent=szBuffer;
    strUserAgent.MakeUpper();
    if(strUserAgent.Find("MSIE")==-1)
        return SF_STATUS_REQ_NEXT_NOTIFICATION;

    dwSize=1024;
    pCtxt->GetServerVariable("SERVER_NAME", szBuffer, &dwSize);
    strServer=szBuffer;

```

```

dwSize=1024;
pCtxt->GetServerVariable("SERVER_PORT",szBuffer,&dwSize);
strPort=szBuffer;

dwSize=1024;
pHeaderInfo->GetHeader(pCtxt->m_pFC,"url",szBuffer,&dwSize);
strURL=szBuffer;
strURL.MakeLower();
///TEMPORARY!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
//  if (strPort=="80" && strURL.Find("&xxxxx=xxxxx&")==-1)
//      return SF_STATUS_REQ_NEXT_NOTIFICATION;
///TEMPORARY!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
CString str="/dualproject/_scriptlibrary/pm.js";
if((pos=strURL.Find(str))!=-1)
{
    strURL.SetAt(pos,0);
    LPVOID lp=pCtxt->AllocMem(sizeof(CUserInfoBuffer),0);
    CUserInfoBuffer* pBuf=(CUserInfoBuffer*)lp;
    strcpy(pBuf->m_szUser,"PMUser");
    strcpy(pBuf->m_szPass,"pm_user");
    pCtxt->m_pFC->pFilterContext=lp;
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

if (strPort!="80")
    strServer+=CString(":")+strPort;

if(strURL.Find(".htm")==-1) //add later more possibilities like asp, cfm,
dll, etc
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
if(strURL.Find("/_vti_inf.html")!=-1)
    return SF_STATUS_REQ_NEXT_NOTIFICATION;

if((pos=strURL.Find("&xxxxx=xxxxx&"))!=-1)
{
    strURL.SetAt(pos,0);
    LPVOID lp=pCtxt->AllocMem(sizeof(CUserInfoBuffer),0);
    CUserInfoBuffer* pBuf=(CUserInfoBuffer*)lp;
    strcpy(pBuf->m_szUser,"PMUser");
    strcpy(pBuf->m_szPass,"pm_user");
    pCtxt->m_pFC->pFilterContext=lp;
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

if(strURL.Find("dualproject")!=-1)
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
if(strURL.Find("fvfvf=fvfvf")!=-1)
    return SF_STATUS_REQ_NEXT_NOTIFICATION;

if(strURL.Find("adext.dll?";0)!=-1)
{
    //  strScript+=CString("&")+pBuf->m_szRef;
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

```

```

/*      dwSize=1024;
        BOOL bRef1=pHeaderInfo->GetHeader(pCtxt->m_pFC, "referer:", szBuffer, &dwSize);
        strReferer=szBuffer;
        strReferer.MakeLower();
        int len=strReferer.GetLength();
        if ((pos=strReferer.Find("adext"))!=-1)
        {
            strReferer.MakeReverse();
            if ((pos=strReferer.Find("&"))!=-1)
            {
                strReferer.MakeReverse();
                strReferer.SetAt(len-pos-1, 0);
                strReferer.MakeReverse();
            }
            if ((pos=strReferer.Find("&"))!=-1)
            {
                len=strReferer.GetLength();
                strReferer.MakeReverse();
                strReferer.SetAt(pos, 0);
                strReferer.MakeReverse();
                BOOL b1=pHeaderInfo->SetHeader(pCtxt->m_pFC, "referer:", strReferer.GetBuffer(0));
            }
        }

        }

        CString strURL1;
        if ((pos=strURL.Find("&xxxxx=framexxxx&"))!=-1)
        {
            strURL.SetAt(pos, 0);
        }

        //
        strURL1="http://" + strServer + "/dpbin/AdExt.dll?d&" + "http://" + strServer + strURL + "&0&frame";
        //
        BOOL b1=pHeaderInfo->SetHeader(pCtxt->m_pFC, "url", strURL1.GetBuffer(0));
        //
        strURL1.ReleaseBuffer();
        return SF_STATUS_REQ_NEXT_NOTIFICATION;
    }

    dwSize=1024;
    BOOL bRef=pHeaderInfo->GetHeader(pCtxt->m_pFC, "referer:", szBuffer, &dwSize);
    strReferer=szBuffer;

    if (bRef)
    {
        if ((pos=strReferer.Find("&xxxxx=xxxxx&"))!=-1)
        {

```

```

        //
        strURL1="http://" + strServer + "/dpbin/AdExt.dll?d&" + "http://" + strServer + str
URL + "&0&";
        //          BOOL b1=pHeaderInfo->SetHeader(pCtxt->
m_pFC, "url", strURL1.GetBuffer(0));
        //          strURL1.ReleaseBuffer();

        return SF_STATUS_REQ_NEXT_NOTIFICATION;
    }
}

//      strURL+=CString("&fvfvf=fvfvf");

        strURL1=CString("http://" + strServer + "/dpbin/AdExt.dll?holder&") + "http://"
+ strServer + strURL + "&fvfvf=fvfvf\r\n\r\n";
        BOOL b1=pHeaderInfo->SetHeader(pCtxt->m_pFC, "url", strURL1.GetBuffer(0));
        strURL1.ReleaseBuffer();
        return SF_STATUS_REQ_NEXT_NOTIFICATION;

//      dwSize=4096;

/*      CString strURL(szBuffer);
        strURL.MakeLower();
        if (strURL.Find("adext.dll?d&") != -1)
        {
            strURL+=CString("&") +
        } */
//      pHeaderInfo->GetHeader(pCtxt->m_pFC, "version", szBuffer, &dwSize);
//      strServer=szBuffer;
//      dwSize=4096;

//      pCtxt->GetServerVariable("SERVER_NAME", szBuffer, &dwSize);

        dwSize=1024;
        BOOL b=pHeaderInfo->GetHeader(pCtxt->m_pFC, "url", szBuffer, &dwSize);
        if(!b)
            return SF_STATUS_REQ_NEXT_NOTIFICATION;

/*      LPVOID lp=pCtxt->AllocMem(sizeof(CAdFBuffer), 0);
        CAdFBuffer* pBuf=(CAdFBuffer*)lp;
        strcpy(pBuf->m_szRef, szBuffer);
        pCtxt->m_pFC->pFilterContext=lp; */

```

```

        return SF_STATUS_REQ_NEXT_NOTIFICATION;
    }

DWORD CAdFilterFilter::OnAuthentication(CHttpFilterContext* pCtxt,
    PHTTP_FILTER_AUTHENT pAuthent)
{
    CUserInfoBuffer* pBuf=(CUserInfoBuffer*)pCtxt->m_pFC->pFilterContext;
    if (pBuf)
    {
        strcpy(pAuthent->pszPassword,pBuf->m_szPass);
        strcpy(pAuthent->pszUser,pBuf->m_szUser);
    }

    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

DWORD CAdFilterFilter::OnUrlMap(CHttpFilterContext* pCtxt,
    PHTTP_FILTER_URL_MAP pMapInfo)
{
    // char szBuffer[4096];
    // DWORD dwSize=4096;

    /* CString strMethod,strScript,strQuery,strPath,strServer,strReferer;
    pCtxt->GetServerVariable("SCRIPT_NAME",szBuffer,&dwSize);
    strScript=szBuffer;
    dwSize=4096;
    pCtxt->GetServerVariable("REQUEST_METHOD",szBuffer,&dwSize);
    strMethod=szBuffer;
    dwSize=4096;
    pCtxt->GetServerVariable("SERVER_NAME",szBuffer,&dwSize);
    strServer=szBuffer;
    dwSize=4096;

    pCtxt->GetServerVariable("QUERY_STRING",szBuffer,&dwSize);
    strQuery=szBuffer;
    strMethod.MakeUpper();
    if(strMethod!="GET")
        return SF_STATUS_REQ_NEXT_NOTIFICATION;    */
    /* strScript.MakeUpper();
    if((pos=strScript.Find("ADEXT?D&",0))!=-1)
    {
        strScript+=CString("&")+pBuf->m_szRef;
        return SF_STATUS_REQ_NEXT_NOTIFICATION;
    } */

    /* strPath=pMapInfo->pszURL;
    strPath.MakeLower();
    if(strPath.Find(".htm")==-1) //add later more possibilities like asp,
    cfm, dll, etc
        return SF_STATUS_REQ_NEXT_NOTIFICATION;    */
    int pos;
    CString strPhysPath(pMapInfo->pszPhysicalPath);
    if((pos=strPhysPath.Find("&xxxxx="))!=-1)

```

```

    {
        //      strPhysPath.Replace("&xxxxx=xxxxx&","");    //this is sign of file
        load, don't call loader
        pMapInfo->pszPhysicalPath[pos]=0;
        return SF_STATUS_REQ_NEXT_NOTIFICATION;
    }
/*    if(strPath.Find("dualproject")!=-1)
        return SF_STATUS_REQ_NEXT_NOTIFICATION;
    if (strQuery.Find("fvfvf=fvfvf")!=-1)
        return SF_STATUS_REQ_NEXT_NOTIFICATION;    */

//    pCtxt->GetServerVariable("HTTP_REFERER",szBuffer,&dwSize);
//    strReferer=szBuffer;
//    dwSize=4096;

//    CString strURL;
/*    CAdFBuffer* pBuf=(CAdFBuffer*)pCtxt->m_pFC->pFilterContext;
    if (pBuf)
    {
        strReferer=pBuf->m_szRef;

        if ((pos=strReferer.Find("&xxxxx=xxxxx&"))!=-1)
        {
            strURL="http://" + strServer + "/dpbin/AdExt.dll?d&" + strScript + "&0&\r\n\r\n";
        }
    }    */

//
//    strURL="http://" + strServer + "/dualproject/holder.asp?file=" + "http://" + strS
server + strScript + "?fvfvf=fvfvf\r\n\r\n";
//
//    strURL="http://" + strServer + "/dpbin/AdExt.dll?holder&" + "http://" + strServer
+ strScript + "&" + pMapInfo->pszPhysicalPath + "&fvfvf=fvfvf\r\n\r\n";
//
//    strURL="http://" + strServer + "/dpbin/AdExt.dll?holder&" + "http://" + strServer
+ strScript + "&fvfvf=fvfvf\r\n\r\n";
//    RedirectURL(pCtxt, strURL);
//    return SF_STATUS_REQ_NEXT_NOTIFICATION;

}

DWORD CAdFilterFilter::OnSendRawData(CHttpFilterContext* pCtxt,
PHTTP_FILTER_RAW_DATA pRawData)
{
    LPTSTR pstrIn;

    pstrIn = (LPTSTR) pRawData->pvInData;

    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

```

```

DWORD CAdFilterFilter::OnReadRawData(CHttpFilterContext* pCtxt,
    PHTTP_FILTER_RAW_DATA pRawData)
{
    LPTSTR pstrIn;
    pstrIn = (LPTSTR) pRawData->pvInData;
    // AfxMessageBox(pstrIn);
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

DWORD CAdFilterFilter::OnLog(CHttpFilterContext *pCtxt, PHTTP_FILTER_LOG pLog)
{
    // TODO: React to this notification accordingly and
    // return the appropriate status code
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

DWORD CAdFilterFilter::OnEndOfNetSession(CHttpFilterContext* pCtxt)
{
    // TODO: React to this notification accordingly and
    // return the appropriate status code
    return SF_STATUS_REQ_NEXT_NOTIFICATION;
}

void CAdFilterFilter::RedirectURL(CHttpFilterContext *pCtxt, LPCTSTR url)
{
    DWORD size = _tcslen(url);
    CString str(url);
    CString str1;
    str1="Location: "+str;
    // pCtxt->m_bSendHeaders = FALSE;
    pCtxt->m_pFC->ServerSupportFunction(pCtxt->m_pFC,
        SF_REQ_SEND_RESPONSE_HEADER,
        (PVOID) "302
Redirect",
        (DWORD)str1.GetBuffer(1),
        NULL);
    str.ReleaseBuffer();
}

// Do not edit the following lines, which are needed by ClassWizard.
#if 0
BEGIN_MESSAGE_MAP(CAdFilterFilter, CHttpFilter)
    //{{AFX_MSG_MAP(CAdFilterFilter)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
#endif
// 0

////////////////////////////////////
// If your extension will not use MFC, you'll need this code to make
// sure the extension objects can find the resource handle for the
// module. If you convert your extension to not be dependent on MFC,
// remove the comments around the following AfxGetResourceHandle()
// and DllMain() functions, as well as the g_hInstance global.

/****

static HINSTANCE g_hInstance;

```



```
HINSTANCE AFXISAPI AfxGetResourceHandle()
{
    return g_hInstance;
}

BOOL WINAPI DllMain(HINSTANCE hInst, ULONG ulReason,
                    LPVOID lpReserved)
{
    if (ulReason == DLL_PROCESS_ATTACH)
    {
        g_hInstance = hInst;
    }

    return TRUE;
}

****/
```

---

Appendix 9 c      Adfilter.def

; ADFILTER.def : declares the module parameters for the DLL.

LIBRARY            "ADFILTER"

EXPORTS

    HttpFilterProc  
    GetFilterVersion

## Appendix 9 d Adfilter.dsp

```

# Microsoft Developer Studio Project File - Name="AdFilter" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=AdFilter - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "AdFilter.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "AdFilter.mak" CFG="AdFilter - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "AdFilter - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "AdFilter - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe

RSC=rc.exe

!IF "$(CFG)" == "AdFilter - Win32 Release"

# PROP BASE Use_MFC 2
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""

```

```

# PROP Use_MFC 2
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "Release"
# PROP Intermediate_Dir "Release"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_WINDLL" /D "_AFXDLL" /Yu"stdafx.h" /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_WINDLL" /D "_AFXDLL" /D "_MBCS" /D "_USRDLL" /Yu"stdafx.h" /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG" /d "_AFXDLL"
# ADD RSC /l 0x409 /d "NDEBUG" /d "_AFXDLL"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 /nologo /subsystem:windows /dll /machine:I386

!ELSEIF "$(CFG)" == "AdFilter - Win32 Debug"

# PROP BASE Use_MFC 2
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 2
# PROP Use_Debug_Libraries 1
# PROP Output_Dir "Debug"
# PROP Intermediate_Dir "Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /D "_WINDLL" /D "_AFXDLL" /Yu"stdafx.h" /FD /GZ /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/D "_WINDLL" /D "_AFXDLL" /D "_MBCS" /D "_USRDLL" /FR /Yu"stdafx.h" /FD /GZ /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG" /d "_AFXDLL"
# ADD RSC /l 0x409 /d "_DEBUG" /d "_AFXDLL"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD LINK32 /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept

!ENDIF

# Begin Target

# Name "AdFilter - Win32 Release"
# Name "AdFilter - Win32 Debug"
# Begin Group "Source Files"

```

```
# PROP Default_Filter "cpp;c;cxx;rc;def;r;odl;idl;hpj;bat"
# Begin Source File

SOURCE=.\AdFilter.cpp
# End Source File
# Begin Source File

SOURCE=.\AdFilter.def
# End Source File
# Begin Source File

SOURCE=.\AdFilter.rc
# End Source File
# Begin Source File

SOURCE=.\StdAfx.cpp
# ADD CPP /Yc"stdafx.h"
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "h;hpp;hxx;hm;inl"
# Begin Source File

SOURCE=.\AdFilter.h
# End Source File
# Begin Source File

SOURCE=.\Resource.h
# End Source File
# Begin Source File

SOURCE=.\StdAfx.h
# End Source File
# End Group
# Begin Group "Resource Files"

# PROP Default_Filter "ico;cur;bmp;dlg;rc2;rct;bin;rgs;gif;jpg;jpeg;jpe"
# Begin Source File

SOURCE=.\AdFilter.rc2
# End Source File
# End Group
# End Target
# End Project
```

---

## Appendix 9 e Adfilter.h

```

#if !defined(AFX_ADFILTER_H_C7D4DB6C_D1CF_11D3_8324_00400517BDC1__INCLUDED_)
#define AFX_ADFILTER_H_C7D4DB6C_D1CF_11D3_8324_00400517BDC1__INCLUDED_

// ADFILTER.H - Header file for your Internet Server
// AdFilter Filter

#include "resource.h"

class CAdFilterFilter : public CHttpFilter
{
public:
    CAdFilterFilter();
    ~CAdFilterFilter();

// Overrides
    // ClassWizard generated virtual function overrides
    // NOTE - the ClassWizard will add and remove member functions
here.
    // DO NOT EDIT what you see in these blocks of generated code !
    //{AFX_VIRTUAL(CAdFilterFilter)
public:
    virtual BOOL GetFilterVersion(PHTTP_FILTER_VERSION pVer);
    virtual DWORD OnPreprocHeaders(CHttpFilterContext* pCtxt,
PHTTP_FILTER_PREPROC_HEADERS pHeaderInfo);
    virtual DWORD OnAuthentication(CHttpFilterContext* pCtxt,
PHTTP_FILTER_AUTHENT pAuthent);
    virtual DWORD OnUrlMap(CHttpFilterContext* pCtxt, PHTTP_FILTER_URL_MAP
pMapInfo);
    virtual DWORD OnSendRawData(CHttpFilterContext* pCtxt,
PHTTP_FILTER_RAW_DATA pRawData);
    virtual DWORD OnReadRawData(CHttpFilterContext* pCtxt,
PHTTP_FILTER_RAW_DATA pRawData);
    virtual DWORD OnLog(CHttpFilterContext* pfc, PHTTP_FILTER_LOG pLog);
    virtual DWORD OnEndOfNetSession(CHttpFilterContext* pCtxt);
    //}AFX_VIRTUAL
    void RedirectURL(CHttpFilterContext *pCtxt, LPCTSTR url);
    //{AFX_MSG(CAdFilterFilter)

```

```

    ///}}AFX_MSG
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before
// the previous line.

#endif //
!defined(AFX_ADFILTER_H__C7D4DB6C_D1CF_11D3_8324_00400517BDC1__INCLUDED)

```

# Appendix 9 f Adfilter.rc

```

//Microsoft Visual C++ generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//

#include "afxres.h"

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

#ifdef APSTUDIO_INVOKED
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include \"afxres.h\"\\r\\n"
    "\\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "#define _AFX_NO_SPLITTER_RESOURCES\\r\\n"
    "#define _AFX_NO_OLE_RESOURCES\\r\\n"
    "#define _AFX_NO_TRACKER_RESOURCES\\r\\n"
    "#define _AFX_NO_PROPERTY_RESOURCES\\r\\n"

```

```

"\r\n"
"\r\n"
#ifdef AFX_RESOURCE_DLL || defined(AFX_TARG_ENU)\r\n"
#ifdef _WIN32\r\n"
LANGUAGE 9, 1\r\n"
#pragma code_page(1252)\r\n"
#endif // _WIN32\r\n"
#include "AdFilter.rc2" // non-Microsoft Visual C++ edited
resources\r\n"
#include "afxres.rc" // Standard components\r\n"
#include "afxisapi.rc" // Internet Support resources\r\n"
#endif\r\n"
"\0"

END
#endif // APSTUDIO_INVOKED

////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "KazaGur\0"
            VALUE "FileDescription", "AdFilter Internet Server Extension
Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "ADFILTER\0"
            VALUE "LegalCopyright", "Copyright (C) 2000 KazaGur\0"
            VALUE "LegalTrademarks", "\0"
            VALUE "OriginalFilename", "ADFILTER.DLL\0"
            VALUE "ProductName", "AdFilter Internet Server Extension\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END
END

```



```
////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDS_FILTER        "AdFilter Filter"
END

#ifndef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//

#define _AFX_NO_SPLITTER_RESOURCES
#define _AFX_NO_OLE_RESOURCES
#define _AFX_NO_TRACKER_RESOURCES
#define _AFX_NO_PROPERTY_RESOURCES

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE 9, 1
#pragma code_page(1252)
#endif // _WIN32
#include "AdFilter.rc2" // non-Microsoft Visual C++ edited resources
#include "afxres.rc"    // Standard components
#include "afxisapi.rc"  // Internet Support resources
#endif

#endif // not APSTUDIO_INVOKED
```

Appendix 9 g      Adfilter.rc2

```
//  
// AdFilter.RC2 - resources Microsoft Visual C++ does not edit directly  
//
```

```
#ifndef APSTUDIO_INVOKED  
    #error this file is not editable by Microsoft Visual C++  
#endif //APSTUDIO_INVOKED
```

```
////////////////////////////////////  
// Add manually edited resources here...  
////////////////////////////////////
```

## Appendix 9 h Resource.h

```
//{{NO_DEPENDENCIES}}  
// Microsoft Visual C++ generated include file.  
// Used by AdFilter.rc  
//
```

---

```
#define IDS_FILTER          101  
  
#define _APS_NEXT_RESOURCE_VALUE  103  
#define _APS_NEXT_CONTROL_VALUE   103  
#define _APS_NEXT_SYMED_VALUE     103  
#define _APS_NEXT_COMMAND_VALUE   32768
```

## Appendix 9 i StdAfx.cpp

```
// stdafx.cpp : source file that includes just the standard includes
// AdFilter.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"
```

## Appendix 9 j StdAfx.h

```
#if !defined(AFX_STDAFX_H__C7D4DB72_D1CF_11D3_8324_00400517BDC1__INCLUDED_)
#define AFX_STDAFX_H__C7D4DB72_D1CF_11D3_8324_00400517BDC1__INCLUDED_

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#include <afx.h>
#include <afxwin.h>
#include <afxmt.h>           // for synchronization objects
#include <afxext.h>
#include <AFXINET.H>
#include <afxisapi.h>

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before
the previous line.

#endif //
!defined(AFX_STDAFX_H__C7D4DB72_D1CF_11D3_8324_00400517BDC1__INCLUDED)
```

## Appendix 10 Adsys.php

```

<%
include((isset($_SERVER[DOCUMENT_ROOT])?$_SERVER[DOCUMENT_ROOT]:di
rname(dirname(dirname($_SERVER[PATH_TRANSLATED]))))
"/sofcast/tools/config.php"); %>
<%
include("$DOCUMENT_ROOT/sofcast/bin/siteinfo.php");
include("$DOCUMENT_ROOT/sofcast/bin/adpicker.php");

header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
header("Cache-Control: max-age=1, no-cache");

$agent = $_HTTP_USER_AGENT;
$isie = !(strpos($agent, "MSIE") === false);
$isphp = !(strpos($agent, "PHP") === false);
$ismoz4 = !(strpos($agent, "Mozilla/4") === false);
$ismoz5 = !(strpos($agent, "Mozilla/5") === false);
$ismoz = $ismoz4 || $ismoz5;
if ($ismoz4 && !$isie) return;

$tmp = $url;
if (!isset($url)) $tmp = $_HTTP_REFERER;
if (list($new,$sid,$enablehold,$frameside) = siteinfo($tmp)) {
    $gotbsite = true;
    //$url = $new;
    $sql = "select siteactive from sites where sid=$sid";
    $res = mysql_query($sql, $connection);
    list($active) = mysql_fetch_row($res);
} else {
    $gotbsite = false;
}

```

```

$server = "$HTTP_HOST";

$adurl = "http://$server/sofcast/bin/adview.php?url=$url";

$frm = 0;
if ($frameside == "R") $frm += 17;
if (!$isie) {
    $siz = adsize($sid, $url, "S");
    if (!$siz === false) {
        $frm += $siz;
    }
}

$frm1 = $frm; $nam1 = "ad"; $url1 = $adurl;
$frm2 = "*"; $nam2 = "content"; $url2 = $url;
$frm3 = "0";
if ($ismo4 && !$isie) $frm3 = "1";
if ($frameside == "R") {
    $frm1 = "*"; $nam1 = "content"; $url1 = $url;
    $frm2 = $frm; $nam2 = "ad"; $url2 = $adurl;
}

%>

document.domain = '<%= $domain %>';
function fixcontent() {
    //alert("onload handler");
    //alert("top = " + top.location.href);
    //alert("con = " + top.content.location.href);
    if (top.content.location.href != top.location.href)
        top.content.location.href = top.location.href;
    //alert("con = " + top.content.location.href);
}

function pmtonclick(e) {
    <% if ($isie) { %>
        if (!window.event) window.event = e;
        var obj = window.event.srcElement;
        var tag = obj.nodeName;
        while (tag != "A" && tag != "AREA") {
            obj =
(obj.parentElement)?obj.parentElement:obj.parentNode;
            if (!obj) return true;
            if (!obj.nodeName) {
                continue;
            }
            var tag = obj.nodeName;

```

```

        alert("tag=" + tag);
    }
    url = obj.href
    if (url == "") return;
    if (url.indexOf("sofcast") > -1) {
        window.open(url, "new");
    } else {
        top.location = url;
    }
    window.event.returnValue = false;
    return false;
<% } elseif ($ismoz4) { %>
    //alert("mousedown");
    obj = e.target;
    while (obj) {
        url = obj.toString();
        //alert("obj='" + url + "'");
        if (url.substring(0,5) == "javas") return;
        if (url.substring(0,5) == "http:") break;
        obj = obj.parentNode;
        if (!obj) return;
    }
    if (url.indexOf("sofcast") > -1) {
        window.open(url, "new");
    } else {
        top.location = url;
    }
    return false;
<% } elseif ($ismoz5) { %>
    //alert("mousedown");
    obj = e.target;
    tag = obj.tagName;
    while (tag != "A" && tag != "AREA") {
        obj = obj.parentNode;
        tag = obj.tagName;
        //alert("tagName='" + obj.tagName + "'
obj.toString()=' ' + obj.toString() + ' ');
    }
    url = obj.toString();
    if (url.indexOf("sofcast") > -1) {
        window.open(url, "new");
    } else {
        top.location = url;
    }
    return false;
<% } %>
}

```



```

var pmtoffset = 0;

function pmthscroll(dist) {
    window.scrollBy(dist,0);
    pmtsize();
    return false;
}

<%
    if ($isie) {
        include("$DOCUMENT_ROOT/sofcast/bin/scroll_ie.php");
    } else {
        include("$DOCUMENT_ROOT/sofcast/bin/scroll_ns.php");
    }
%>

if (window == top) {
    ag=navigator.userAgent.toLowerCase();
    mv=parseInt(navigator.appVersion);
    isie=((ag.indexOf("msie") != -1) && (ag.indexOf("opera") == -
1));
    loaded = 0;
    holdad = 0;
    sframe = 1;
    document.write('<frameset onLoad="fixcontent();" border=1
cols="<%= $frml%>,<%= $frmc%>,<%= $frmr%>">');
    document.write('<frame src="<%= $urll%>" name="<%= $naml%>"
scrolling="no" noresize>');
    document.write('<frame src="<%= $urlc%>" name="<%= $namc%>"
scrolling="yes" noresize>');
    document.write('<frame
src="http://<%= $server%>/sofcast/bin/blank.html" name="daemon">');
    document.write('<noframes>');
} else {
<% if ($isie) { %>
    document.onclick = pmtonclick;
<% } else { %>
    document.captureEvents(Event.MOUSEDOWN);
    document.onmousedown = pmtonclick;
<% } %>
    document.write("<div id=pmthscr name=pmthscr style='position:
absolute; top: 0; left: 0; visibility: hidden;'>");
    document.write("<table cellpadding=0 cellspacing=0 width=100%
bgcolor=#e4e4e4>");
    document.write("<tr>");
    document.write("<td><a onclick='return pmthscroll(-5);'>");

```

```
document.write("<img id=pmtlbut border=0 width=14 height=14  
src='http://<%= $server %>/sofcast/bin/arrowl.gif'>");  
document.write("</td>");  
document.write("<td id=pmthwin width=100%>");  
document.write("<div id=pmthbut name=pmthbut style='position:  
absolute;'>");  
document.write("<img  
src='http://<%= $server %>/sofcast/bin/arrowt.gif' width=14  
height=14 border=0 ondragstart='return false;'>");  
document.write("</div>");  
document.write("</td>");  
document.write("<td><a onclick='return pmthscroll(5);'>");  
document.write("<img id=pmtrbut border=0 width=14 height=14  
src='http://<%= $server %>/sofcast/bin/arrowr.gif'>");  
document.write("</td>");  
document.write("<tr>");  
document.write("</tr>");  
document.write("</table>");  
document.write("</div>");  
  
pmtGrabHandlers();  
setTimeout("pmtGrabHandlers();", 500);  
  
}
```

## CLAIMS

1. A method of delivering content and advertisement, the method comprising:
  - receiving a request for a target web page;
  - sending, in response to the request for the target web page, a frames page comprising a plurality of frames and a frames page program module, wherein the plurality of frames comprises a content frame and an advertisement frame;
  - receiving a request to load the content frame;
  - sending content to load the content frame, wherein the content comprises the target web page;

---

  - receiving a request to load the advertisement frame; and
  - sending an advertisement to load the advertisement frame.
2. The method of claim 1 further comprising:
  - determining whether the request for the target web page is a first request during a current session.
3. The method of claim 1, wherein the request to load the content frame and the request to load the advertisement frame are prompted by the frames page program module.
4. The method of claim 1 further comprising:
  - retrieving a list of advertisements from an advertisement table; and
  - selecting the advertisement from the list.
5. The method of claim 4, wherein said selecting the advertisement comprises randomly selecting the advertisement from the list.

6. The method of claim 1, wherein the plurality of frames further comprises a daemon frame.

7. The method of claim 6, wherein the daemon frame is initially loaded with a blank page.

8. The method of claim 6, wherein the daemon frame is loaded with server-side scripted content.

9. The method of claim 8 further comprising receiving a request to insert advertising event information into a tracking database.

10. The method of claim 1 further comprising:  
modifying the content by adding a content page program module to the content, wherein the content page program module comprises software for performing the following function:  
handling mouse click events.

11. The method of claim 10, wherein the content page program module further comprises software for performing the following functions:  
handling window OnLoad events; and  
handling window OnBeforeUnload events.

12. The method of claim 1 further comprising:  
modifying the advertisement by adding an advertisement page program module to the advertisement, wherein the advertisement page program module comprises software for performing the following functions:

displaying HOLD and HOME buttons; and  
handling button click events for the HOLD and HOME buttons.

13. The method of claim 1, wherein the frames page program module comprises software for performing the following functions:

loading content; and  
loading advertisement.

14. The method of claim 13, wherein the frames page program module further comprises software for performing the following functions:

handling hyperlink mouse click actions; and  
allocating screen area.

15. The method of claim 1, wherein the frames page program module comprises software for causing display of the advertisement frame concurrently with the content frame.

16. The method of claim 1, wherein the frames page program module comprises software for causing display of the advertisement frame between, in terms of time, display of two successive content pages.

17. The method of claim 16, wherein the frames page program module comprises software for causing the advertisement frame, loaded with the advertisement, to be stored in a browser cache while the content is displayed.

18. The method of claim 17, wherein the frames page program module comprises software for causing the advertisement frame, loaded with the advertisement, to be displayed when a user requests to view another content page.

19. The method of claim 16, wherein, in an interstitial mode, prior to a hyperlink click action by a user, the advertisement loaded in the advertisement frame is not viewable by the user, further wherein upon the hyperlink click action by the user, the frame is displayed instantly to the user.

20. A method of delivering content and advertisement, the method comprising:

receiving a request for a target web page;

sending, in response to the request for the target web page, a web page that causes, directly or indirectly, establishing a frames page comprising a plurality of frames and a frames page program module, wherein the plurality of frames comprises a content frame and an advertisement frame;

receiving a request to load the content frame, wherein the request to load the content frame is prompted by the frames page program module;

sending content to load the content frame, wherein the content comprises the target web page;

receiving a request to load the advertisement frame, wherein the request to load the advertisement frame is prompted by the frames page program module; and

sending an advertisement to load the advertisement frame.

21. The method of claim 20, wherein the web page includes code that invokes a program, wherein execution of the program by a browser causes establishing the frames page.

22. The method of claim 20, wherein the web page comprises a program, wherein execution of the program by a browser causes establishing the frames page and the plurality of empty frames.

23. The method of claim 20, wherein the plurality of frames further comprises a daemon frame.

24. The method of claim 20, wherein the frames page program module comprises software for causing display of the advertisement frame along with the content frame.

25. The method of claim 20, wherein the frames page program module comprises software for causing display of the advertisement frame between display of successive content pages.

---

26. The method of claim 25, wherein the frames page program module comprises software for causing the advertisement frame, loaded with the advertisement, to be stored in a browser cache while the content is displayed.

27. The method of claim 26, wherein the frames page program module comprises software for causing the advertisement frame, loaded with the advertisement, to be displayed when a user requests to view another content page.

28. A method of delivering content and advertisement, the method comprising:

receiving a request for a target web page; and

sending, in response to the request, a modified target web page, wherein a hyperlink in the modified target web page has a corresponding hyperlink in the target web page, wherein the corresponding hyperlink references a first server, further wherein a request using the hyperlink in the modified target web page is directed to a server other than the first server, further wherein the modified target web page

includes code that, upon a hyperlink click on the modified target web page, invokes functionality of a frames page program module to resize a content frame and an advertisement frame.

29. The method of claim 28, wherein the target web page is modified by rewriting the corresponding hyperlink to a rewritten hyperlink, wherein the rewritten hyperlink references software modules at the server.

30. The method of claim 28, wherein the target web page is modified by adding thereto code that intercepts the hyperlink click on the modified target web page, further wherein the code directs requests based on the hyperlink click to the server.

31. The method of claim 30, wherein the code invokes a program, wherein execution of the program by the browser causes establishing a frames page comprising a plurality of frames and the frames page program module, wherein the plurality of frames comprises the content frame and the advertisement frame.

32. The method of claim 30, wherein the plurality of frames further comprises a daemon frame.



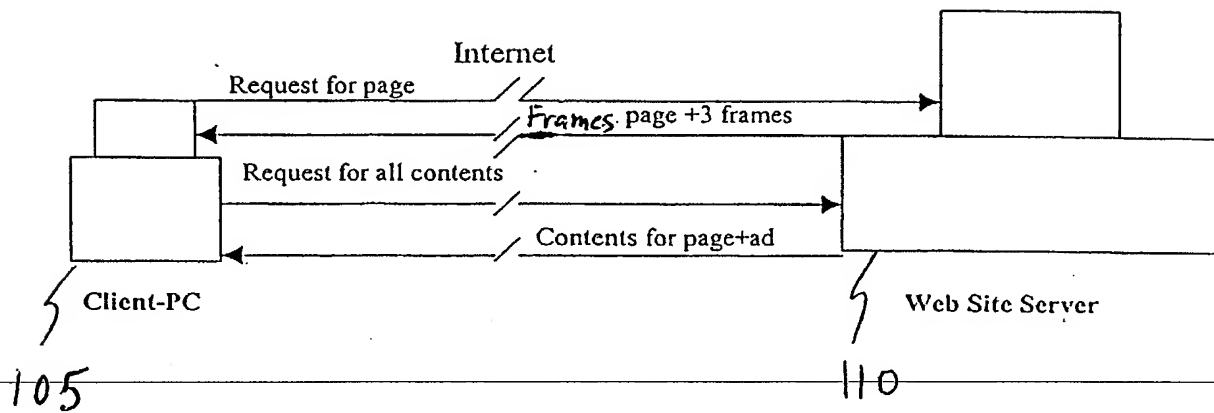


Figure 1

2/10

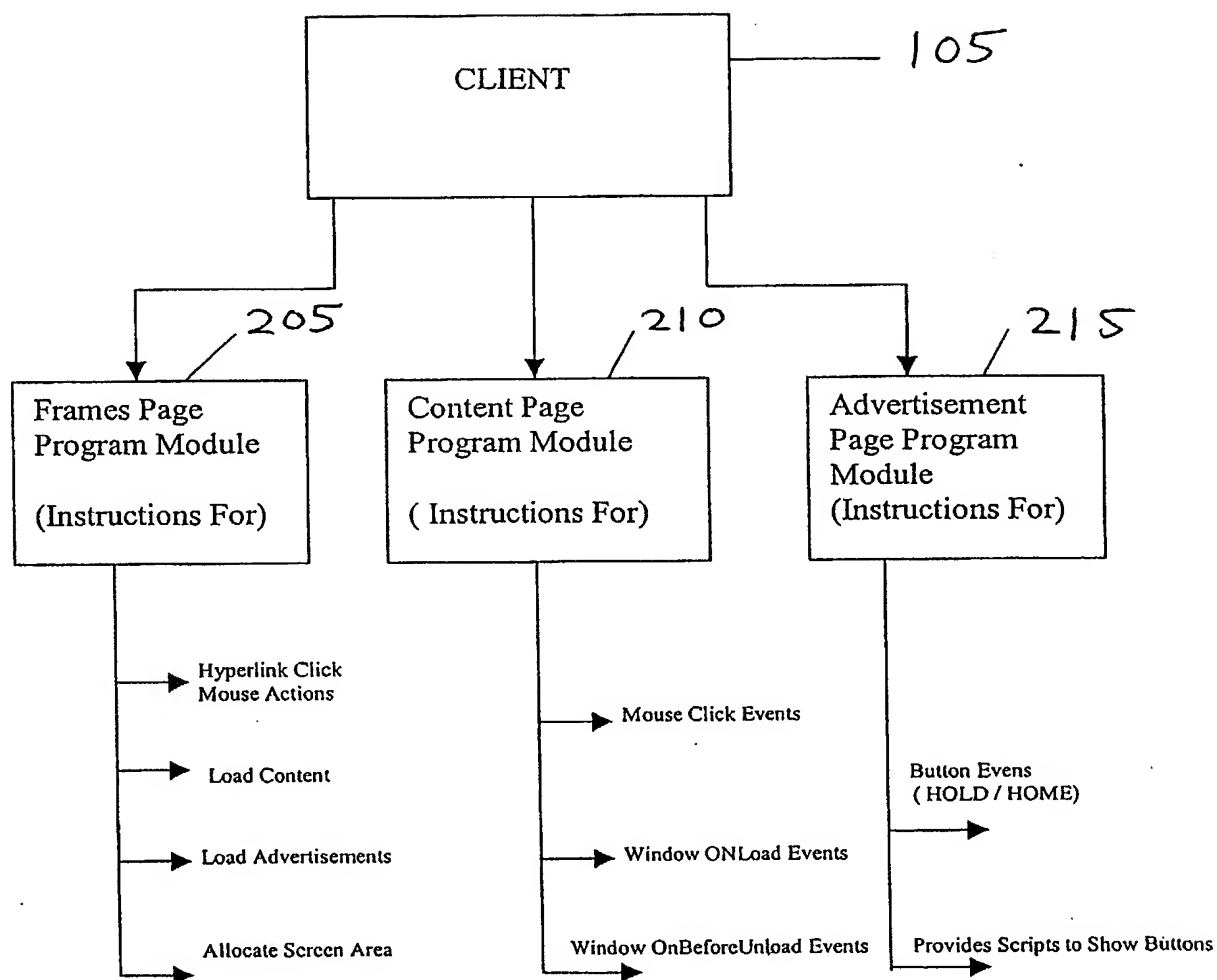


Figure 2

3/10

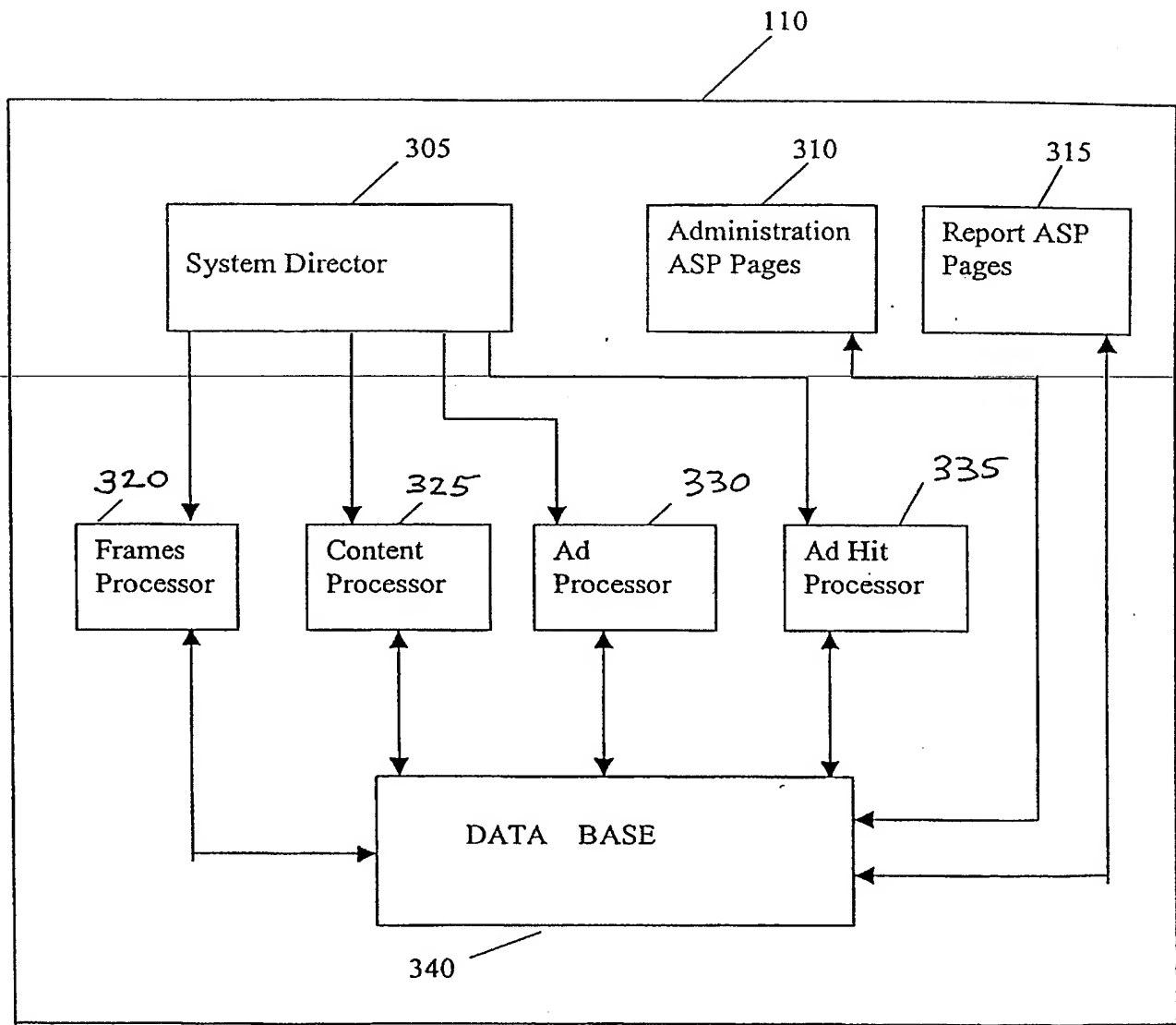


Figure 3

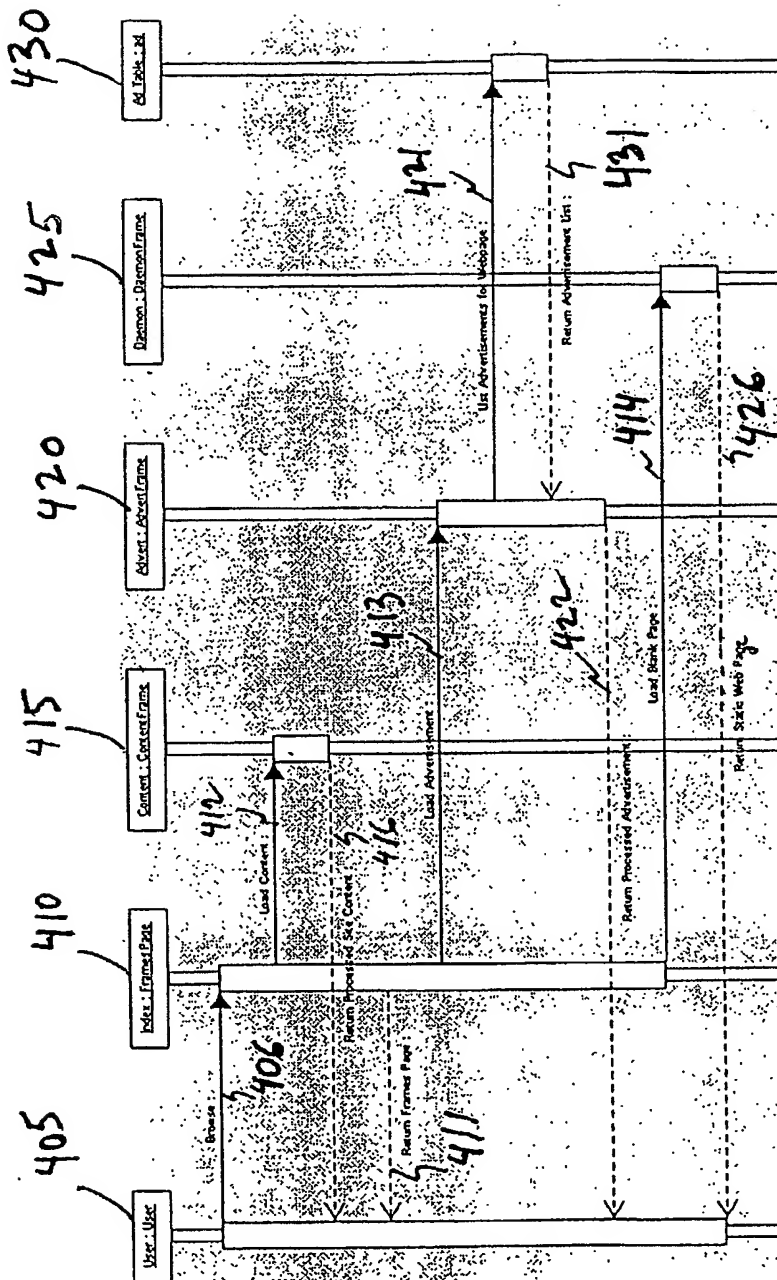


Figure 4

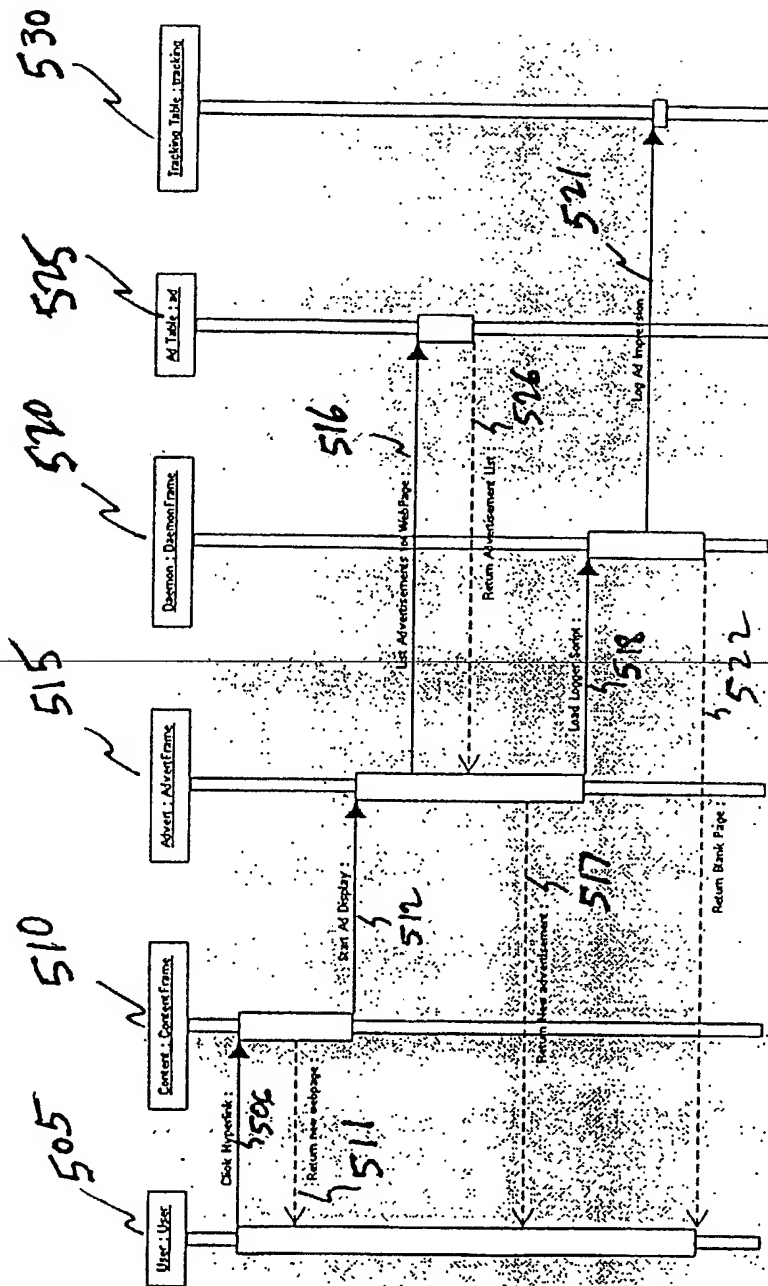


Figure 5

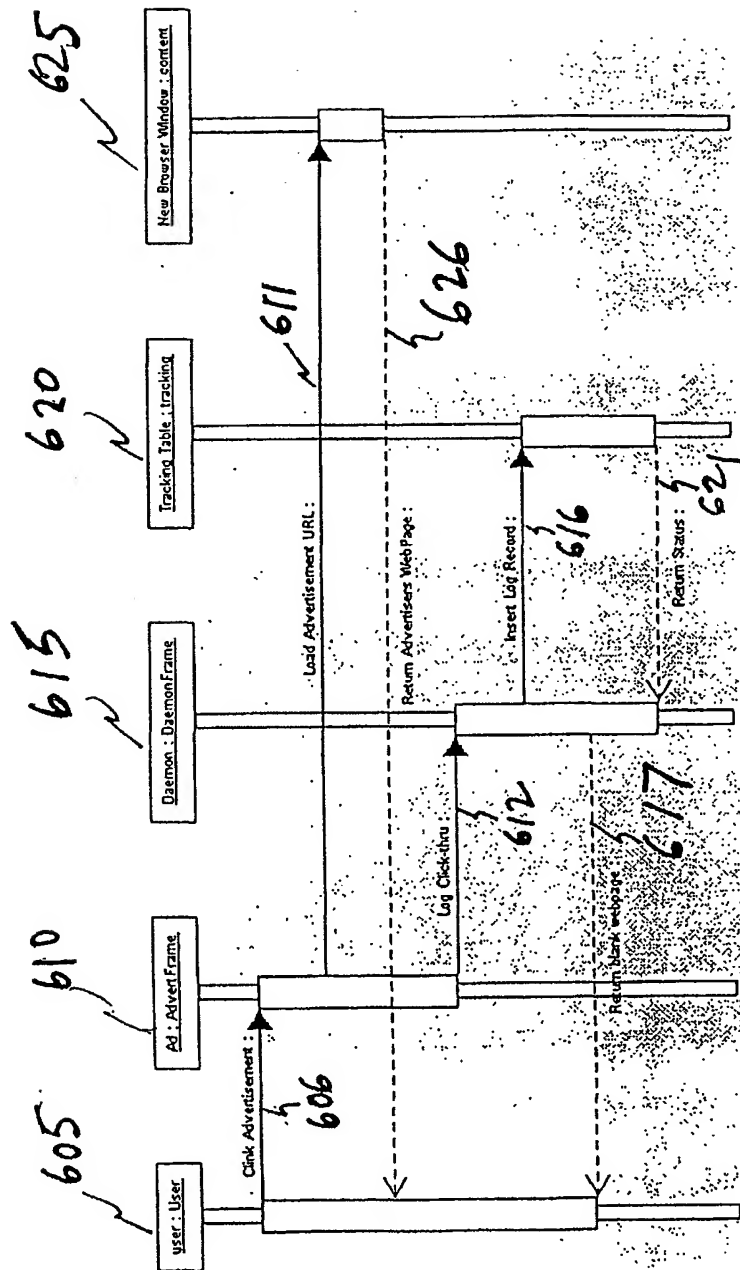


Figure 6

7/10

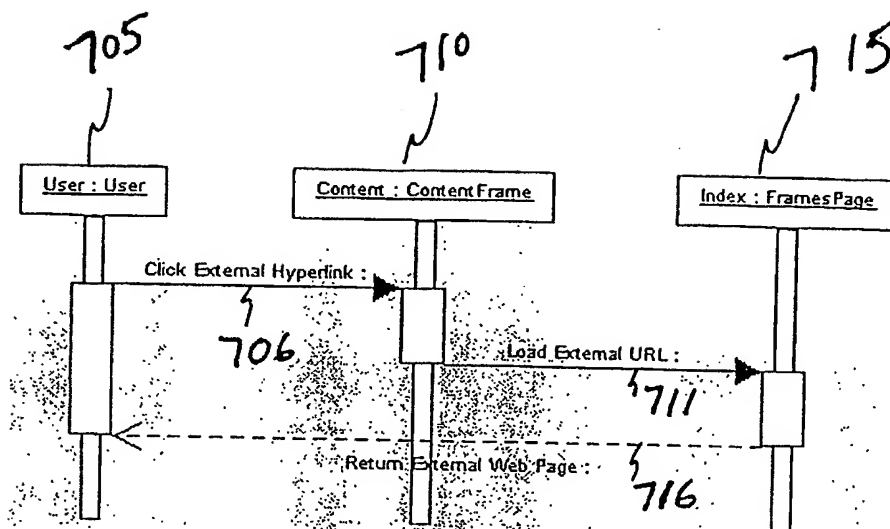


Figure 7

8/10

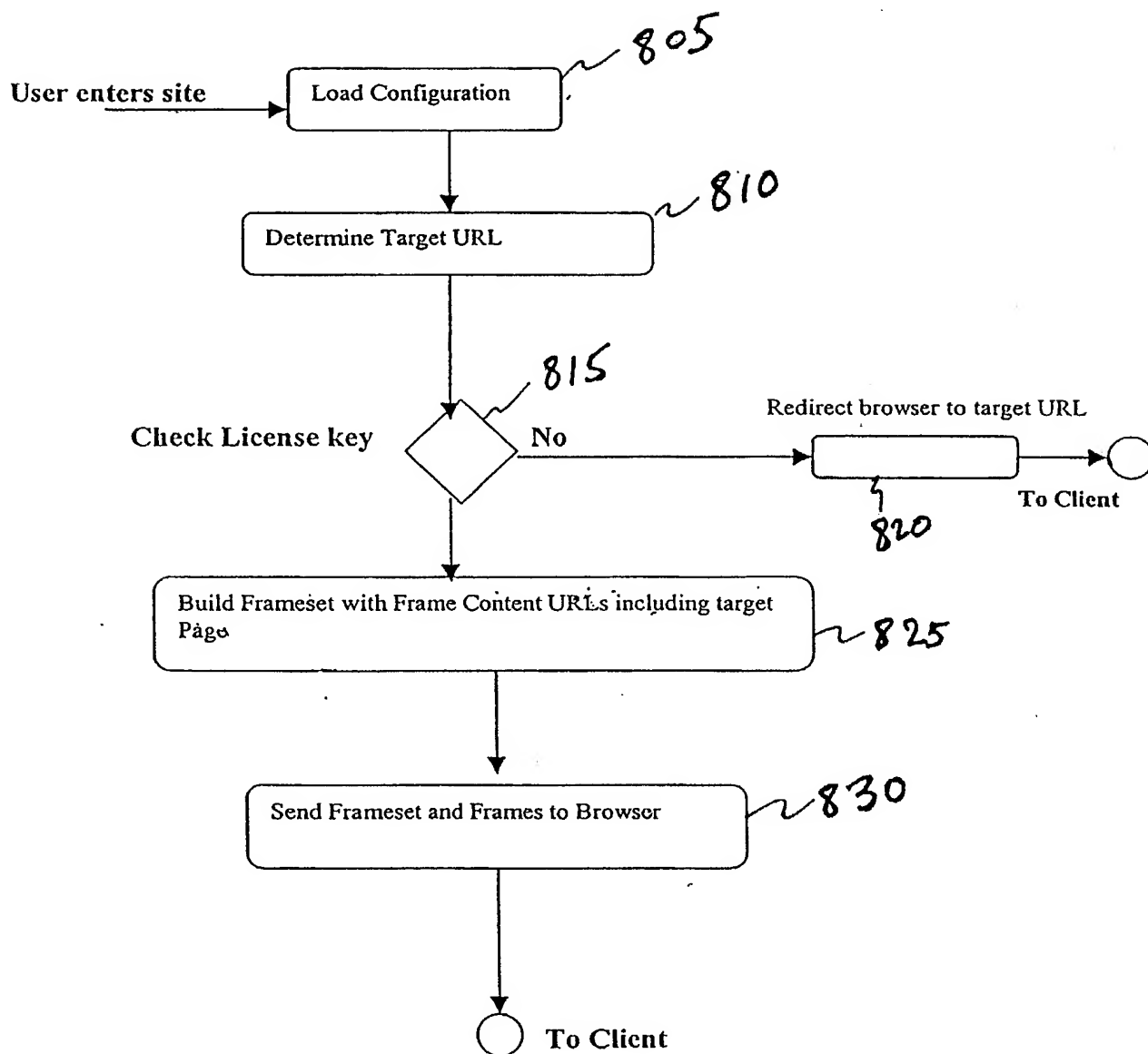


Figure 8



9/10

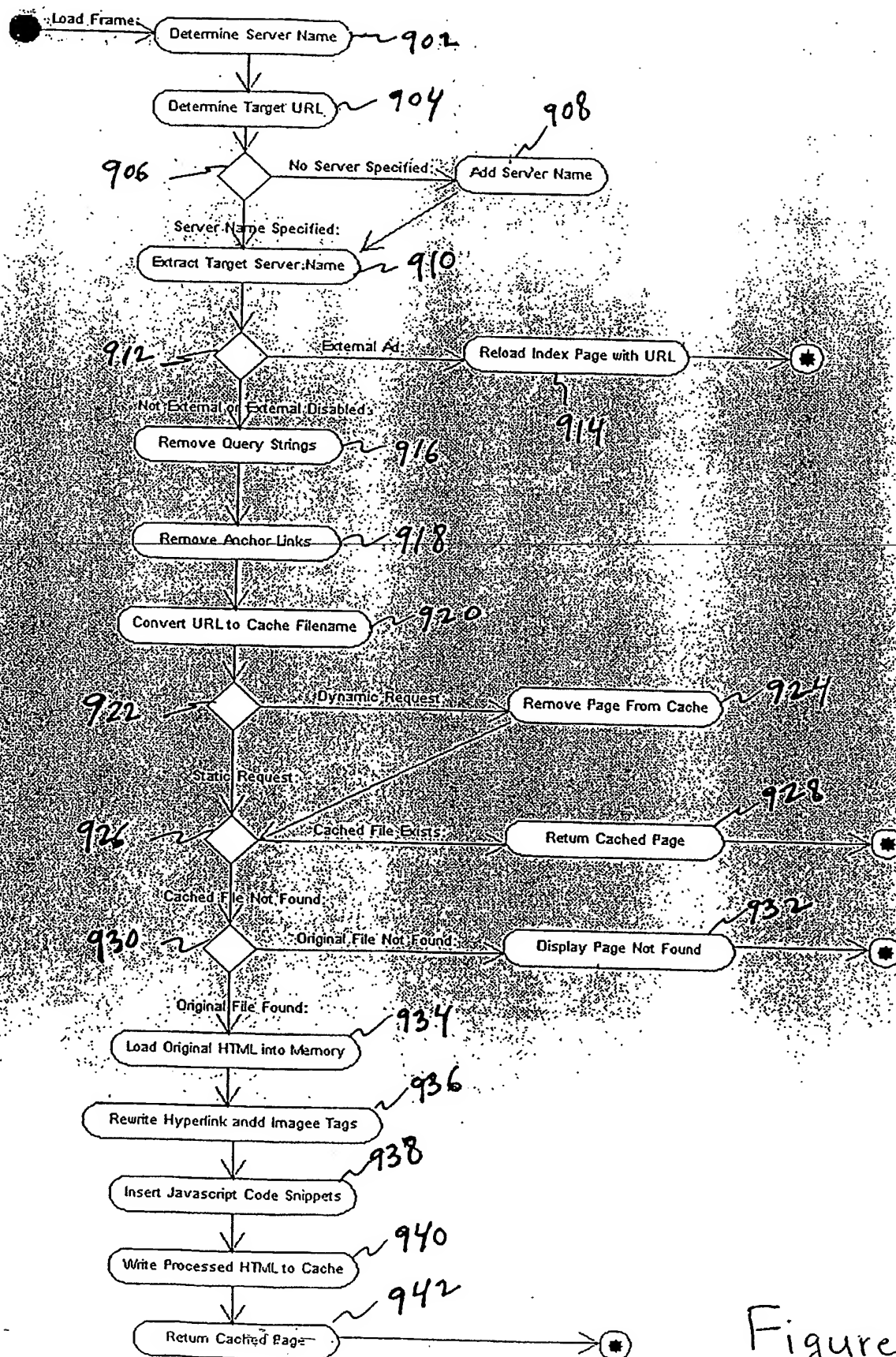


Figure 9

10/10

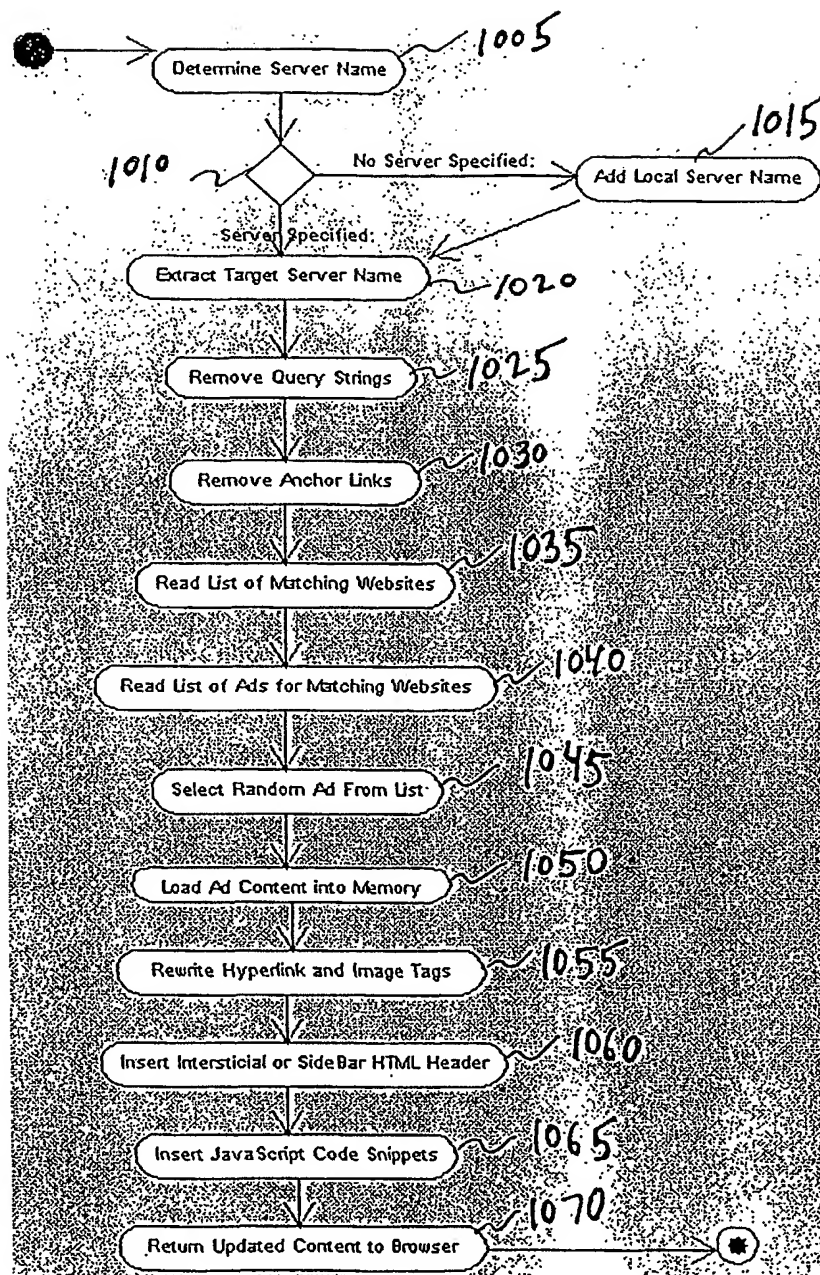


Figure 10

# INTERNATIONAL SEARCH REPORT

International application No.

PCT/US02/39288

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/60; G06F 3/00, G06F 13/38

US CL : 705/14; 709/203, 219, 225

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/14; 709/203, 219, 225

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,913,040 A (RAKAVY ET AL) 15 June 1999 (15.06.1999), whole document	1-32
Y	US 6,314,451A (LANDSMAN et al.) 6 November 2001 (06.11.2001), whole document.	1-32
Y	US 6,268,856 B1 (BRUCK et al.) 31 July 2001 (31.07.2001), whole document.	1-32
Y	US 6,011,537 A (SLOTZNICK) 4 January 2000 (04.01.2000), whole document.	1-32
A	US 5,740,549 (REILLY et al.) 14 April 1998 (14.04.1998), whole document	1-32

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:	
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

17 April 2003 (17.04.2003)

Date of mailing of the international search report

29 APR 2003

Name and mailing address of the ISA/US

Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703)305-3230

Authorized officer

Yuhanga Retia

Telephone No. (703) 308-1113